VRIJE
UNIVERSITEIT
BRUSSEL

Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Applied Sciences and Engineering: Computer Science

EDITABLE AND CUSTOMIZABLE KNOWLEDGE MAPS

Thijs Spinoy

June 2019

Promoter:
Prof. Dr. Olga De Troyer

Advisor:
Jan Maushagen

Sciences and Bio-Engineering Sciences

VRIJE
UNIVERSITEIT
BRUSSEL

Masterproef ingediend in gedeeltelijke vervulling van de eisen voor het behalen van de graad Master of Science in de Ingenieurswetenschappen: Computerwetenschappen

EDITABLE AND CUSTOMIZABLE KNOWLEDGE MAPS

Thijs Spinoy

Juni 2019

Promoter:
Prof. Dr. Olga De Troyer

Assistent:
Jan Maushagen

Wetenschappen & Bio-ingenieurswetenschappen

Abstract

Many tools nowadays support the creation of visualizations of linked data, e.g. mind map tools. Unfortunately, most of these tools lack other important functionality, like being able to define your own look and feel for your visualization, to create templates, or to do more than just creating a mind map summarizing your thoughts. This thesis presents *GuideaMaps 2.0*, a browser-based tool to create knowledge maps in a convenient way and which does not lack the previously mentioned functionality. With convenient we mean: not being limited to a particular device or platform and being able to use the tool for several purposes. While a first version of GuideaMaps was mainly created for the purpose of requirement elicitation, we broadened the range of situations in which the tool can be used. As an example use case, a website with a complicated tree structure underneath was visualized by our tool to make its structure more clear. In a user study with 52 participants evaluating this use case we found our visualization was easier to use than the original website.

Another contribution is that the system is created as a library, i.e. developers can extend and modify the implementation of the tool if this is needed for their goal. The library allows to customize the visualization of the nodes and the links without affecting the default implementation of the tool. A custom implementation can easily be plugged in. Further, two modes are foreseen: (1) map creators can create templates for a specific goal and (2) end-users can fill the templates with the necessary data. Hence, GuideaMaps 2.0 is different than the existing tools in many ways. Other tools are often usable for a single purpose, while our application tries to be functional in many situations.

The standard GuideaMaps visualization was evaluated with another user study. The results showed that creating such a tool is not straightforward. Small details (e.g. icons) can lead to frustration and make the tool less intuitive.

Samenvatting

Vandaag de dag bestaan heel wat tools om verbanden tussen data te visualiseren, e.g. mind map tools. Helaas missen de meeste van deze tools belangrijke functionaliteit, zoals de mogelijkheid om je eigen "look and feel" voor je visualisatie te definiëren, om templates te maken, of om meer te kunnen dan enkel een mind map maken die je gedachten ordent. In deze thesis wordt GuideaMaps 2.0 voorgesteld, een browser-gebaseerde tool om op een eenvoudige manier kennismappen te maken en die de reeds genoemde functionaliteit niet mist. Met "eenvoudig" bedoelen we: niet gelimiteerd zijn tot een bepaald apparaat of platform en over de mogelijkheid beschikken om de tool voor meerdere doeleinden te kunnen gebruiken. Terwijl de eerste versie van GuideaMaps voornamelijk gemaakt werd voor requirement elicitation hebben we het spectrum van situaties waarin de tool gebruikt kan worden verbreed. In een illustrerende use case werd een website met een ingewikkelde boomstructuur gevisualiseerd door onze tool om de structuur duidelijker te maken. In een user study met 52 deelnemers werd het duidelijk dat onze visualisatie eenvoudiger was in gebruik in vergelijking met de website.

Een andere belangrijke bijdrage is dat het systeem gemaakt is als een library, i.e. ontwikkelaars kunnen de implementatie van de tool uitbreiden en bewerken als dit nodig is voor het doel dat ze voor ogen hebben. De library laat toe om de visualisatie van de nodes en de links aan te passen zonder de standaard implementatie van de tool te beïnvloeden. Zo een eigen visualisatie kan eenvoudig ingeplugd worden. Verder zijn er twee modi voorzien: (1) map creators kunnen templates maken voor een bepaald doel en (2) eindgebruikers kunnen de templates vullen met de nodige data. We kunnen dus zeggen dat GuideaMaps 2.0 op veel vlakken verschillend is van bestaande tools. Andere systemen zijn vaak bruikbaar voor één enkel doel, terwijl onze applicatie probeert om functioneel te zijn in veel verschillende situaties.

De standaard GuideaMaps visualisatie werd geëvalueerd in een andere user study. De resultaten toonden aan dat zo een tool maken niet triviaal is. Kleine details (e.g. symbolen) kunnen leiden tot frustraties en de tool minder intuïtief maken.

Acknowledgements

In this section, I would like to express my appreciation to the people who contributed directly or indirectly to the result of this thesis.

First of all, I want to say a heartfelt thank you to professor Olga De Troyer for being my promoter, but also for proofreading this thesis many times, being there every time I needed advice or had questions, answering my numerous e-mails very quickly, ... the whole year. Next to my promoter, my assistant during this thesis, Jan Maushagen, also deserves a big thank you for all the time and effort he has put in giving me tips and tricks about the implementation.

Further, I would like to thank César Callebaut and the schools he contacted to find participants for the user study.

Last but not least, a special and important group of people needs to be thanked as well: my family. From my parents for giving me the chance and faith to study Computer Science for five years, over my sister and in-laws to my fantastic girlfriend, who I can always count on. Dank u Yanake!

Declaration of Originality

I hereby declare that this thesis was entirely my own work and that any additional sources of information have been duly cited.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this thesis has not been submitted for a higher degree to any other University or Institution.

Contents

List of Figures

Listings

1 Introduction

The more people can store in their memory, the less they have to look up in external sources and the faster they can work. However, processing and storing a big amount of information in the human memory for later reuse is not easy, especially when the information is presented in the form of a long text. Therefore, students for example, make schemes and summaries of their study material. The reason why they do this is to make it easier to learn and remember the material. Schemes and visualizations make relations between pieces of information explicit and are easier to grasp. In addition, most people are visual learners, meaning that they learn better when using visual aids. Not only the way of learning new subject material, but also other activities can be supported by means of a visualization. For instance, if you write down the structure of a computer program in words, it is more difficult to discuss that structure with other people than when the structure is expressed by means of a diagram. This can be explained as follows.

According to Moody (2009), there exists a difference between visual notations and textual languages in how they encode information and how they are processed by the human mind: textual representations are one-dimensional and are processed serially by the auditory system, while visual representations are two-dimensional and processed in parallel by the visual system.

Further, the form of representations/visualizations has a greater effect on understanding and problem solving than their content (Moody, 2009). In other words, for a visualization, the way the content is represented is more important than the content itself.

Moody (2009) also defines the term *Cognitive effectiveness* as "the speed, ease and accuracy with which a representation can be processed by the human mind".

The better the cognitive effectiveness of a visual notation, the better human communication and problem solving can be done.

The goal of this thesis is to create a tool with high cognitive effectiveness, in which it is possible to represent linked data and knowledge in a visual manner. It should not only be possible to create a visual notation of the data, but also to edit the data as well as extending the representation with additional data. The solution is based on a visualization tool created by Janssens (2013), called GuideaMaps. This application was mainly built to provide support for the requirement elicitation for serious games, i.e. (digital) games used for purposes other than mere entertainment (Susi et al., 2007). It provides the functionality to enter information in pre-defined maps (trees of nodes). More details about this application can be found in chapter 2. We present a new version of GuideaMaps, which can be used for other purposes than serious games and which has a bunch of interesting improvements under the hood.

1.1 Problem Statement

For the first version of GuideaMaps, the main goal was to develop the following:

> "A tool that allows the different people (and with different background) involved in the development of a serious game (e.g., against cyber bullying) to brood over the goals, characteristics and main principles of a new to develop serious game. The tool should be easy to use and usable in meetings. Therefore, we want to explore the characteristics and capabilities of a tablet (i.e. iPad)." (Janssens, 2013)

By specifying the goal in this way, end users of the application are restricted in different ways. First, they need an iPad to be able to use the application. Another type of tablet with a different operating system is not possible, because GuideaMaps was created and designed for iOS only. If someone doesn't have access to an iPad, (s)he cannot use the application, which is a hard restriction.

Furthermore, the tool focuses on requirement elicitation. Initially, the tool was created for the purpose of requirement elicitation for serious games, but it can also be used for the requirement elicitation in other domains (De Troyer & Janssens, 2014). However, because the visualizations are based on pre-defined templates, the nodes can only be edited by the end-user in a limited way: content can be given and the background color can be changed, but the end-user cannot add new nodes. In addition, the visual notation used is fixed: the creators of the visualization templates cannot edit the representation of a node or define their own representation (e.g. change the length and width or use a different shape). Not being able to do this is a limitation in the sense that for some purposes this default visualization may not be very suitable. Furthermore, for defining a template the author had to use XML and no graphical editor was available for this purpose making it harder for non-ICT schooled people to define new templates.

1.2 Research Goals

The issues discussed in the previous section indicate that the first version of GuideaMaps comes along with some limitations. Therefore, the following research goals for a new version of GuideaMaps were formulated:

Goal 1
The new version of GuideaMaps should work on all common devices and on different operating systems.

Goal 2
It should be possible to pre-define the maps, i.e. the templates, in a graphical way.

Goal 3
The new version of GuideaMaps should allow the end-user to extend and modify the pre-defined map in some restricted way.

Goal 4
The application should be generic in such a way that it can be customized to be usable for different purposes, i.e. the user should be able to define its own graphical representation for the visualization.

This thesis presents a new version of GuideaMaps, called *GuideaMaps 2.0*, taking the research goals into account. How the tool achieved the formulated research goals is explained into detail in the rest of this thesis.

1.3 Thesis Structure

In the next chapter, we start with a brief explanation of the first version of GuideaMaps. Because our solution is mainly based on this application, we start that chapter with a discussion of the different concepts and principles used. We advise the reader to go through this chapter if (s)he is not familiar with GuideaMaps as the rest of this thesis will refer to these concepts and principles.

Chapter 3, called "Related Work", discusses different visualization techniques and existing tools for similar purposes.

Chapter 4 defines the different user classes for the system and the main requirements for the system divided into three categories, i.e. functional, usability and other requirements.

These two chapters are followed by a chapter on the implementation of our tool. Information about choices made and which technologies are used can be found in this chapter. Furthermore, we explain how GuideaMaps is implemented and how the code is structured as a library so that this library can also be used for other

purposes.

In chapter 6, we demonstrate the use of the library for another use case, demonstrating the achievement of Research Goal 4.

The work developed has also been evaluated by means of two user studies. The approach and the results of the evaluation can be found in chapter 7.

In the last chapter, the conclusions of this thesis are presented as well as possible future work.

2

Background: GuideaMaps

This chapter provides a description of the first version of the tool. We present the system and explain its limitations to have a good view of the elements we certainly want to improve in our new version. The information is based on the master thesis of Erik Janssens (Janssens, 2013), the related publication (De Troyer & Janssens, 2014), and the last version of the tool.

2.1 Goals & Principles

GuideaMaps is an iOS-based iPad application based on mind maps. It can be seen as an ideation tool, i.e. a tool that mimics the thinking pattern that people follow when engaged in inventive thinking, e.g. about new products or systems (Goldenberg et al., 1999). The name "GuideaMaps" is a concatenation of "Guided" and "Ideas" and each node in the map is called a "Guidea". The initial goal of the tool was to provide support for the requirement elicitation process for serious games (Janssens, 2013). Later on the tool was extended to also support requirement elicitation for other domains. The only prerequisite is that the requirement elicitation process of the domain is already well known, i.e. it is known which sort of requirements need to be specified for the system in the domain. Therefore, it is called domain-specific requirement elicitation. Example domains are serious games and e-commerce systems.

The issues, i.e. requirements, to consider for a particular domain during requirement elicitation are modeled by means of a kind of predefined mind map, called a *Guidea template*. Such a template provides the different issues to consider, explains them, indicates when issues are required and which are optional, provides possible options and alternatives, indicates the impacts of choices, and allows

documenting everything.

A Guidea template is sub sequentially used to guide the requirement elicitation process of a new system in the given domain. This is done by selecting or deselecting optional issues, selecting possible options, and providing explanations for all decisions made. The result is an actual *GuideaMap*.

In the visualization, the main issue, i.e. idea, is positioned in the center, while all sub-ideas are placed around it. Each idea is represented as a node (called Guidea). An example is shown in Figure 2.1. Mandatory Guideas are linked to their parent by a solid arrow line, while non-mandatory Guideas are linked with a dotted arrow line.

The user of a Guidea template cannot change the structure of the map but he can change some aspects of the layout. To allow the user to group related nodes in a visual way, (s)he can change their color. A user can choose to change the color only for the selected node or for the selected node and all its children. Another way to group nodes is by drag and drop, i.e. the user can change the position of the nodes (using drag and drop) until (s)he obtains a configuration that is more suited for the data or the purpose. These principles support the similarity and proximity principle of the Gestalt Psychology Theory (Koffka, 2013)

2.1.1 Nodes

The regular visualization of a node is a rectangle with rounded corners with a title and some content. If there is not enough space for the content in the visual representation of the node, only the beginning of it is visible. The user can read and edit the complete content by double tapping the node. On double tap, a modal is opened and the keyboard is made visible on the iPad to let the user read and/or edit the data. Note that initially, i.e. when starting to create a Guidea map from a template, the content shown (in the visualization) in the nodes is the description for the Guideas given by the creator of the template. Once the user added a motivation or some other type of information for the Guidea, then this will be shown in the visualization.

Next to regular Guideas, there is another type of Guideas, i.e. the so-called "choice nodes". These nodes allow the user to select a choice from a predefined set of options. For example, the choice node with title "Gender" can have three possibilities which can be added to the visualization, i.e. Male, Female or X. Each time an option is selected, a regular node appears with the name of the option as title. In the template, it can be specified how many options can be or must be selected by the user. This restriction is specified by means of "cardinalities", i.e. a lower limit and an upper limit for the number of selections. Graphically, these nodes are smaller than the regular ones and only have a title. While it is possible to change the color of regular nodes, choice nodes always have the same color. This difference is made to let them stand out (similarity principle of Gestalt Psychology Theory (Koffka, 2013)).

Each node (a regular as well as a choice node) can be marked as optional. In that case, it is not mandatory to fill the content of the node or to select a choice (for choice nodes). As already indicated, these nodes can be recognized by the the dotted line that links them to their parent. The user can decide to use the node and add some content to it, or he can deselect the node, i.e. not use it. A deselected node becomes grey and is less opaque than the other nodes.

2.1.2 GuideaMaps

As already explained, the user starts a GuideaMap by selecting a template from a list of available templates. He can create different maps from the same template and at any time, the user can switch between GuideaMaps he created and has been working on. It is also possible to delete an existing GuideaMap.

A GuideaMap can be exported by means of email and imported by another user. The email is also providing the content of the GuideaMap in a textual and readable form.

2.2 Limitations

The first version of the tool works very well but has some limitations we would like to overcome in our version of the system.

1. The tool is created for iPad only. Hence, it is not possible to use it on other devices.

2. The structure of a map, i.e. the nodes and the links, is fixed and defined by a template on which the map is based.

3. To define a template, you have to use XML. There is no graphical interface to create templates. This can be a hard task for non-ICT schooled people.

4. The system is not collaborative: people cannot work together and simultaneously on the same GuideaMap.

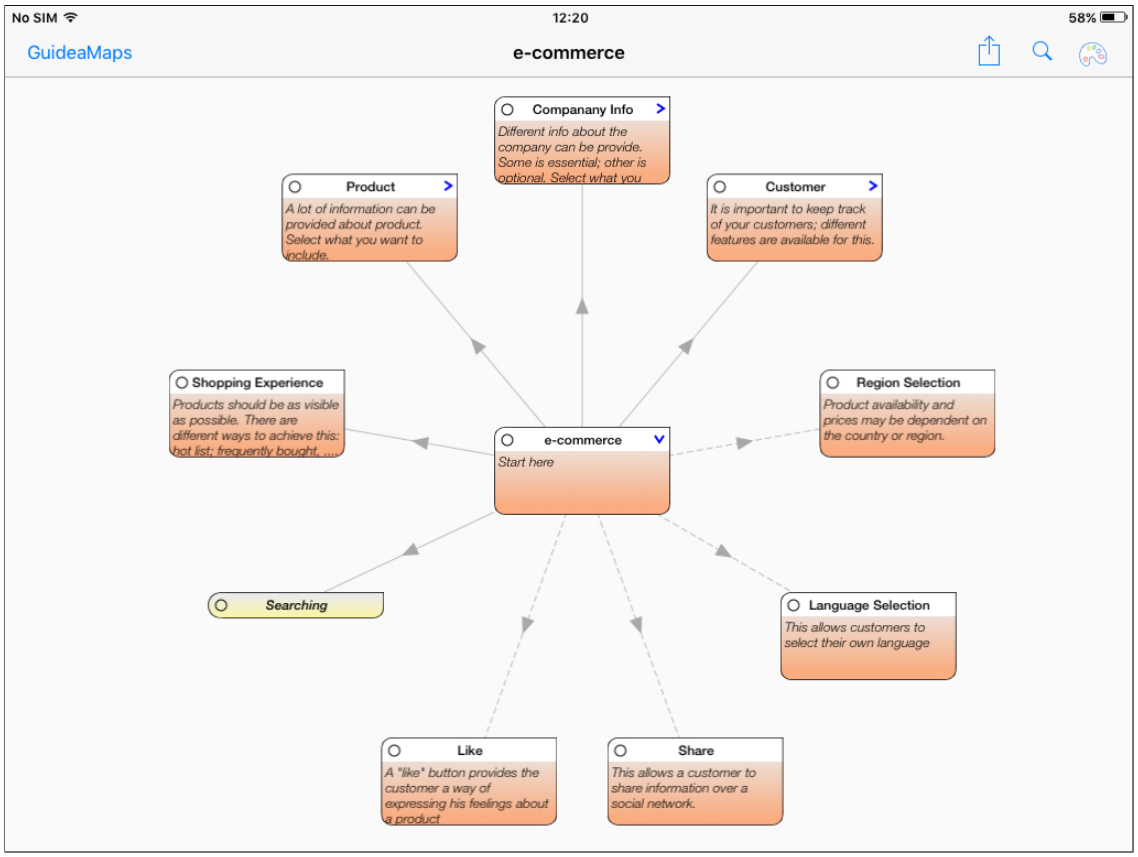


Figure 2.1: E-commerce example created in the original GuideaMaps application.

3 Related Work

There exist lots of ways to visualize data and the relations between data. In this chapter, we discuss visualization techniques related to the needs of GuideaMaps 2.0 (section 3.1). In section 3.2, we discuss related tools.

3.1 Visualization Techniques

3.1.1 Mind Maps

The most well-known technique to visualize related data is to create a mind map (a.k.a. idea map). This technique is mainly used to show the relation between portions of information and for brainstorming purposes. Other applications where this technique is used are note-taking, and problem solving. (Balaid et al., 2016)

In the domain of brainstorming, the goal is to collect and link as many ideas as possible. According to Shih (2011), brainstorming is "an idea generation technique that focuses on using quantity to breed quality". Mind maps are created by writing the main idea in the middle of the drawing, while all sub-ideas are placed around that center node. Each sub-idea is connected with its parent by means of a line. Hence, this kind of visualization is not difficult to create or understand. Even though it can become very large, its simplicity is one of the reasons why it is used a lot in practice.

Because mind maps is not a new concept, but one that most people already know quite well, we will only discuss one important aspect about this visualization technique. When using a digital version of mind maps, in general, the user can change the position of the nodes. Wiegmann et al. (1992) state that maps taking the

Gestalt principles (Koffka, 2013) into account would be more performance-effective than maps that don't integrate Gestalt principles. Therefore, digital mind map systems usually allow their users to re-organize the nodes using drag and drop and by changing the color of the nodes. In this way, the user can, for example, put nodes containing similar data closer to each other and give them the same color (proximity and similarity principle).

3.1.2 Visual Metaphors

Another technique that can be used to represent content is the use of visual metaphors.

> "A visual metaphor is a graphic structure that uses the shape and elements of a familiar natural or manmade artefact or of an easily recognizable activity or story to organize content meaningfully and use the associations with the metaphor to convey additional meaning about the content." (Eppler, 2006)

This technique could be used if you want to help users to memorize the most important elements of a topic, method or concept. In that case, you have to choose a metaphor which has properties in common with the topic, concept or method. (Eppler, 2006) An example of a visual metaphor is shown in Figure 3.1. It shows "a mixed visual metaphor/conceptual diagram template to structure learning content in a narrative structure during a class room discussion" (Eppler, 2006).

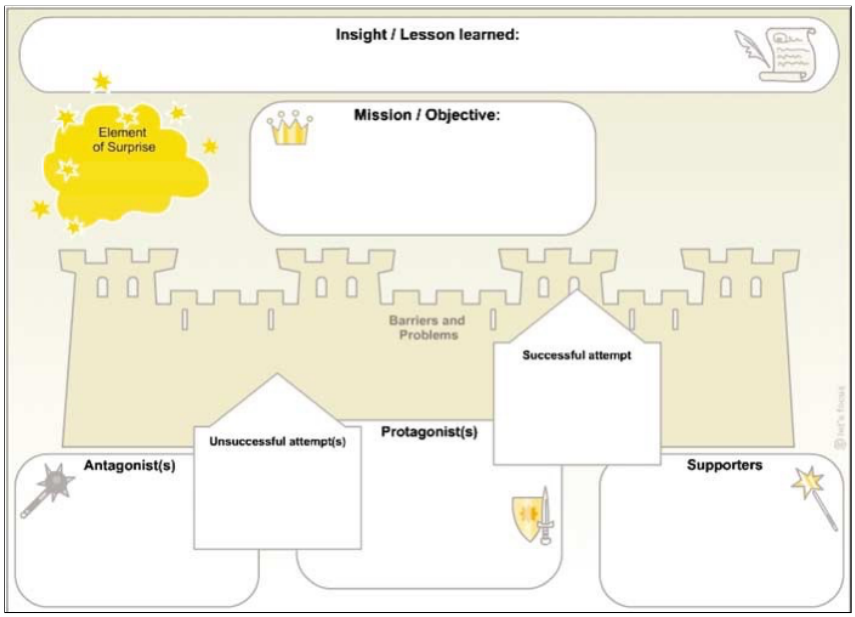


Figure 3.1: Example of a visual metaphor (Eppler, 2006).

As you can see in the figure, the visualization is completely adapted to the specific case of the topic. In this way, it is very difficult to create a generic metaphor, which can be used for different topics. As our tool should be usable for different topics, the technique of visual metaphors is less suitable for the needs of our tool.

3.1.3 Knowledge Maps

According to Balaid et al. (2016), *knowledge maps* is an umbrella term for tools and techniques like mind maps. O'Donnell et al. (2002) defined the concept as follows:

> "Knowledge maps are node-link representations in which ideas are located in nodes and connected to other related ideas through a series of labeled links."

This way of representing information has for example a positive impact on students. The paper of O'Donnell et al. (2002) teaches us that students using knowledge maps are better in remembering the main ideas of the subject in comparison to the ones that study from the text without the visualization. As GuideaMaps's purpose is more focused on representing large amounts of knowledge in an easy to grasp way, remembering the content is less important. However, the node-link representation with the main idea in the center also showed to be useful for this purpose, as illustrated by the popularity of mind maps.

Next to mind maps, concept maps is a second technique included under the umbrella of knowledge maps. Concept maps are in some sense similar to mind maps but they do have some different characteristics. First, the purpose of a mind map is to associate ideas, topics or things, while concept maps illustrate relations between concepts. Further, the structure of a concept map is mostly hierarchical and visualized like a tree. On the other hand, mind maps sometimes have a radial layout and not hierarchical. (Davies, 2011) An analysis by Nesbit & Adesope (2006) showed that concept maps are more effective in grasping knowledge than reading long texts and being present in lectures, which is an analogous result to what O'Donnell et al. (2002) showed us about knowledge maps.

Hence, we can state that GuideaMaps makes use of a knowledge map visualization and more specifically some kind of combination of mind maps and concept maps.

3.2 Existing Tools

In this section, we will discuss several already existing tools with similar functionality as defined in the goals of our application.

3.2.1 Browser-based

To create a digital mind map, a lot of browser-based tools exist yet. They are all very similar and only differ in a small number of functionalities. Therefore, we only discuss two examples of such tools.

Bubbl.us[1] is an online mind-mapping tool with very interesting features. Because it is browser-based, it works on all platforms and devices. Furthermore, you do not have to download any software to be able to use it. The biggest drawback

[1] `https://bubbl.us/`

of the system is that the nodes only contain a title. It is impossible to provide some content in addition to the title of the node. Hence, this tool is probably only useful to write down simple ideas instead of more complex structures of related data. Another drawback is that you can only change the color of the nodes but you cannot customize their shape. But, as a user, you can drag and drop the nodes, share the visualization, work on it simultaneously, etc. Hence, Bubbl.us seems to be a well-created tool with lots of advantages. However, for the purpose of GuideaMaps 2.0, we need some functionality this tool does not provide.

A second browser-based tool is "MindMeister"[2]. It is very easy to create an account on which your different visualizations can be stored. Users are allowed to drag and drop the nodes to other positions and change their layout in terms of color and borders. However, you cannot show a preview of the content in the node itself. You always have to click the node to be able to see the content. Further, it is not possible to customize the links, which can be a drawback because now they are represented as a solid line and it is not possible to add an arrow on them. As a consequence, this is a second well-created tool but missing some important functionality for GuideaMaps 2.0.

3.2.2 Moodle

Scherl et al. (2012) described Moodle as follows:

> "Moodle is a widely used learning management system. It visualizes inner and interdisciplinary relations between learning objects and is generated dynamically depending on user set parameters and interactions. It is a free open-source software package that provides a course management system for the implementation of internet-based learning environments."

In Moodle, navigation between courses only works by binding URLs to words or pictures, similar to the way Wikipedia does it. This way of linking resources makes it difficult to "stay on top of things". Therefore, Scherl et al. (2012) extended Moodle to provide a clear orientation throughout navigation. In this extended version, new learning activities can be linked to already existing learning activities in any course present on the Moodle platform. In so-called concept map-based navigation, the students can open a full-screen concept map (example in Figure 3.2). Every node represents a learning object and a click on it opens the corresponding resource. Further, the currently visited learning object (node) is always in the center of the map, which is dynamically changed and updated based on user interactions.

Hence, this system is comparable with the one we want to build: in our system, the currently selected node will be at the center as well and it should be possible to open nodes to see their content. However, important functionality is missing: similar to the browser-based examples, there is only a title visible for the nodes and no preview of the content. More important is that it is not possible to customize

[2] `https://www.mindmeister.com/`

the layout of the nodes and the links, which is a requirement we certainly want to achieve in our solution.

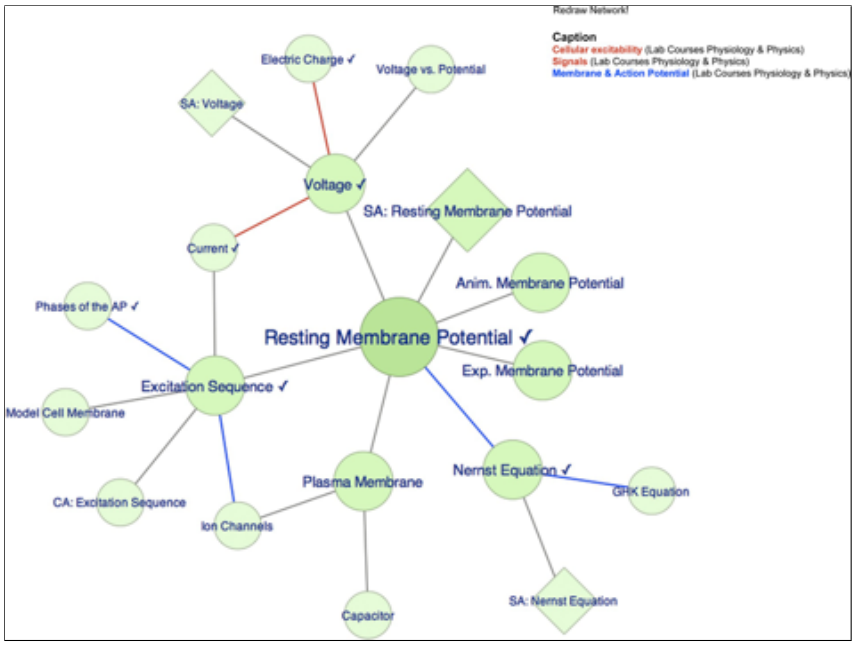


Figure 3.2: Example of concept map-based navigation in Moodle.

3.2.3 CmapTools

Novak & Cañas (2006) did the following:

> "Extending the use of concept mapping to other applications such as the integration of concept mapping with the World Wide Web (WWW) led to the development of software that enhanced the potential of concept mapping, evolving into the current version of CmapTools now used worldwide in schools, universities, corporations, and governmental and non-governmental agencies. CmapTools is a client-server software tool to facilitate the construction and sharing of concept maps."

Cañas et al. (2004) explain that CmapTools allows to publish knowledge models in concept map servers (CmapServers) and that these concept maps can be linked to related concept maps and to other types of media (e.g. images and web pages) in other servers. Further, the tool is also made collaborative. However, our own experience of experimenting with the software was a bit disappointing. The layout of the editor is quite old-fashioned and unsharp and the possibilities are not very clear. It seems new users have to spend some time to get used to the system. For our approach, we want to avoid such a situation, i.e. the possible actions in our application should be intuitive so that users do not lose time searching for the functionality they need.

4

User Classes & Requirements

In general, an application should meet a lot of requirements in order to deliver some quality to the users. A system should provide the right kind of functionality, but it should also be *usable*, i.e. easy to learn and easy to use. Below are two definitions of usability which are often used:

Definition 1

> Usability is a measure of the ease with which a system can be learned and used, its safety, effectiveness and efficiency, and attitude of its users towards it. (Preece et al., 1994)

Definition 2

> Usability is the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use. (ISO, 1998)

Note that the second definition emphasizes the fact that usability is dependent on the target users and on the context of use. This means that one system can be usable for one type of user but not for another type of user or usable in one context but not in another context. Therefore, we also have to consider the different users of a system. In general, in user-centered design methods, the users are classified into user classes. Users in a user class are similar in terms of their characteristics and how they use the system.

Hence, we can distinguish between functional requirements and usability requirements. The system should meet the functional requirements to provide the right functionality and it should meet the usability requirements to be usable. There are also requirements that do not belong to either category, e.g. the so-called non-functional requirements.

This chapter starts by identifying the different user classes. Next, we will present and justify the major requirements formulated for our system, functional as well as usability requirements and other requirements. Detailed requirements and requirements that are rather straightforward are omitted here. Later in this thesis, when we explain the implementation details, we will come back to these requirements and discuss how we managed to meet them.

4.1 User Classes

For our system, we can identify three user classes: (1) developer, (2) map creator and (3) end-user. The class the user belongs to will define its rights and possibilities. An overlap in the rights is possible, i.e. users of different classes can have a number of rights and possibilities in common. The different user classes are described into more detail in the following subsections.

4.1.1 Developer

A developer is a user who can define and implement a custom visualization for the nodes and the links. This means that a developer can make sure that the tool can be used with a different visualization while the possibilities (in terms of functionality) are still the same. Developers are the only users that are able to do this because map creators and end-users are not allowed to extend the code with a custom implementation. The exact steps that need to be taken in order to provide a custom visualization in the tool are explained in chapter 5 and 6.

4.1.2 Map Creator

A map creator has the rights to create a so-called Guidea template or change the structure of an existing template. With the structure, we do not mean the layout of the nodes and the links, but the composition of the map itself. Hence, the map creator "initializes" the map by creating the nodes and links necessary for his goal. He provides a title and a description for each node. As a map creator, it is not possible to already provide the content of the node because that is the responsibility of the end-user. Note that this user corresponds to the template creator in GuideaMaps 1.0. However, in GuideaMaps 1.0 this had to be done in XML, while in our version it will be done in a graphical way to achieve Research Goal 2. We use the term "map" here (instead of template) because the term "map" is easier to understand than the term "template" by non-ICT schooled users.

4.1.3 End User

The end-user is the most restricted user in terms of rights and functionalities. He will use the map defined by a map creator and fill it with content. The end-user cannot change the title or the description, provided by the map creator, of a node but only provides the content. We have this restriction because otherwise this would give them the opportunity to change the structure of the map (which is

a right reserved for the map creator). If the map creator did not provide this information, the end-user can add it, but never change it again afterwards, because, once the information is added, it is the same situation as when the map creator would have provided it.

Further, an end-user is allowed to add child-nodes but they cannot delete nodes, because otherwise it would be possible for end-users to change the pre-defined map completely. The purpose of allowing the end-user to edit the map in some restricted way is to allow them to adjust the map to situations that were not foreseen by the map creator.

4.2 Requirements

4.2.1 Functional Requirements

FR 1. The system needs two modes: an end-user mode and a map creator mode. Depending on the mode used, the corresponding functionality should be available.

FR 2. A map creator should be able to define a structure for the visualization and provide a title and description for each added node and options for choice nodes.

FR 3. Research Goal 3 formulated earlier stated that the new version of GuideaMaps should allow the end-user to extend and modify the pre-defined map in some restricted way, i.e. they should be able to change the background color of nodes, to add missing child nodes, and it should also be possible to add new options in choice nodes.

FR 4. Map creators should be able to delete nodes as well, while end-users can only add missing ones.

FR 5. Users should be able to collapse nodes to hide their children and expand nodes to show their children again.

FR 6. Zooming in or out such that you get less or more information at the same time on the screen is a frequently provided functionality for large visualizations. A feature to zoom is, for example, very useful in situations where you want to compare different parts of the visualization or focus on a certain part.

FR 7. A variation on the zooming feature is "zoom to fit" (a.k.a. zoom until the complete figure fits into the bounding box). With custom zooming, the user can set the zooming level to meet its needs. Zoom to fit automatically adapts the zooming level and moves the content of the application until everything fits on the screen. This feature can, for example, be very useful to get an overview of the current data in the visualization.

FR 8. Research Goal 4 formulated earlier states that the application should be usable for different purposes. While GuideaMaps was usable in the context of domain specific requirement elicitation, our tool should also be useful in many other cases. Therefore, some requirements concerning the genericity of the application are needed:

(a) The tool should be generic in such a way that it is possible to use a different representation, e.g. shape/size for the nodes and the links.

(b) The major implementation should not be changed to achieve (a). An implementation for the nodes and the links created by a developer, should be *plugged in* into the system, without affecting the rest of the implementation.

FR 9. Multiple users should be able to work on the same visualization at the same time.

4.2.2 Usability Requirements

UR 1. The tool should not only be usable for people with experience in Computer Science. It does not matter whether or not the user has a background in Computer Science, he should be able to easily learn to use the system in a short time.

UR 2. Possible actions should be labeled by clear, unmistakable icons. The icons should not be ambiguous, each icon should link to one particular action and thus the user should know exactly what to expect when clicking on the icon. Well chosen icons are one of the factors in the design that contribute to learnability and ease of use.

UR 3. Gestures for common actions should not differ from the gestures used for the same action in other applications. (e.g. the scrolling gesture is probably the best gesture for zooming, because this is a well-known way to zoom in applications (e.g. Google Maps)). Using the same gesture as in other applications will improve the learnability of our tool.

UR 4. The target audience of the application should not be limited in some way: e.g. colorblind people should also be able to use the system.

UR 5. End-users should be able to make small changes to the structure of the map, such as adding extra options to choice nodes, without having to contact the map creator to perform these updates.

4.2.3 Other Requirements

OR 1. To achieve Research Goal 1, the application should run on different kinds of devices (i.e. tablets, laptops, and desktops) and operating systems (i.e. Android, iOS, MacOS, and Windows). The only restriction on the used device is that it needs to have a screen that is large enough because it is

not very convenient to work with the visualization on small screen areas, e.g. on smartphones. The application could run on smartphones but it is not recommended nor required to use it on devices with relatively small screens.

OR 2. The core of the application should be completely separated from custom implementations created by developers.

5 Implementation

The previous sections explained *what* the application should do, which goals should be achieved and which requirements we surely want the application to meet. In this section, more details are provided about *how* all this is translated to an implementation.

5.1 Web Application

Before we started with the actual implementation, we had to decide how to achieve Research Goal 1, i.e. how will we make sure the application is device- and OS-independent? (Requirement OR 1) A first technique could have been to make use of the Java Virtual Machine. This approach would work but we want to make it possible to let multiple people work on the same visualization (FR 9), eventually at the same time. Therefore, it is more convenient to have a browser-based application, i.e. Web Application.

Nowadays, a lot of browsers exist, each with their own characteristics and differences. Because the goal of this thesis was to have a functional prototype, it is not required that the application works perfectly on *all* possible browsers. However, we wanted to make sure that the tool works without any issues in one of the most popular browsers, being Google Chrome.

Another possibility would have been to use a cross-platform development tool like PhoneGap[1] or Cordova[2]. The benefit of those tools is that you can create a web-based application which then is converted to target platforms with simple

[1] `https://phonegap.com/`
[2] `https://cordova.apache.org/`

commands (e.g. “cordova platform add ios” and “cordova build ios”). Hence, from then on the application can be installed on the device itself. A drawback of this is that the app cannot be used if the device has not enough disk space. Further, the app is possibly not up-to-date at any time. This can become an issue when the app would be collaborative and multiple users with different versions of the app want to work with it. However, incompatible versions can be avoided by forcing the user to update to the latest version before he can continue to use the app. Furthermore, cross-platform tools are very well suited for simple UIs, while in the case of more complex UIs, they are not always applicable. Therefore, we concluded that a browser-based solution is better for our goal than the Java Virtual Machine or cross-platform tools.

5.2 Used Technologies

A lot of technologies exist to create Web applications and visualizations. A number of technologies assist in the developing process of such an application. In the following subsections we present the selected technologies, as well as why we chose for this particular technology and not for the alternatives.

5.2.1 ReactJS

ReactJS[3] is an open source library to create user interfaces. One of its main goals is to provide the best possible rendering performances[4]. Performance is good because ReactJS allows developers to break down the user interface into different components. Each component has its own *state*, which contains information about the content of the component. This state can be updated while the application is running and if such an update is made, only this component is re-rendered instead of re-rendering the complete UI[5]. Hence, this provides a huge benefit for the performance. Next to that, it is also not very hard to learn to code in React comparing to other frameworks (e.g. AngularJS). If the developer knows HTML and JavaScript, he will be able to code in ReactJS quickly.

Because of these benefits, ReactJS is the framework in which the GuideaMaps application is implemented. Each node and each link is considered as a separate and unique React Component. The most important reason for implementing the nodes like this is performance: if the state of the node is updated, only this node is re-rendered and not the complete UI.

Alternatives for ReactJS are for example AngularJS and VueJS[6]. ReactJS is by far the most used of these three. Further, it is said to be easier to learn because you only have one structure (“Component”), while in AngularJS this is not the case.

[3] https://reactjs.org/index.html

[4] https://medium.com/@thinkwik/why-reactjs-is-gaining-so-much-popularity-these-days-c3aa686ec0b3

[5] https://facebook.github.io/react/docs/why-react.html

[6] https://medium.com/@TechMagic/reactjs-vs-angular5-vs-vue-js-what-to-choose-in-2018-b91e028fa91d

Also, VueJS lacks resources and is not used a lot in practice, which makes it more difficult to discuss problems with other developers. Maybe the most important benefit of ReactJS is its performance; ReactJS is really fast and our tool should have fast rendering times as well. Hence, because the alternatives seem to have some important drawbacks, we chose for ReactJS as the framework for creating the application.

5.2.2 d3

Another helpful tool is d3. With d3-hierarchy[7], it is possible to transform JSON-data into hierarchical data. Having this kind of data makes it much easier to create a tree- or cluster-structure. In the case of GuideaMaps, a clustered visualization is very useful. The *main*-node (a.k.a. the root node) is then positioned at the center, such that its child nodes can be placed around it. Hence, the further a node is away from the center, the lower it is in the hierarchy. Also, the visualization will not be messed up by positioning the nodes in this way, because every node has exactly one parent. This means there will not be a spaghetti of links where you cannot see from which node the link comes and to which node it is pointing.

Further, d3 makes it easier to create a zoomable layout. The tool is able to update the positions of the nodes each time the zooming level is changed. Hence, as a developer, you do not have to take care of the updated positions if you make use of d3. By using this technique, the functional requirements about zooming (FR 6 and FR 7) are easily achieved.

The alternatives for d3 most of the time come with a problem d3 does not have. For example ChartJS and ChartistJS are limited in the number of features: you can create nice charts with it, but d3 provides more visualizations than only charts. Further, some of the alternatives are commercial (Highcharts, Webix). d3 is open-source and provides functionality to zoom, to create a cluster of nodes based on JSON-data describing these nodes, etc. Because of the wide range of possibilities with d3, the fact that it is widely used and supported by all modern browsers and that it is able to act together with ReactJS, we believe that d3 was a better choice than its alternatives.

5.2.3 Tailwind CSS

Nowadays, the "look and feel" of an application is very important. Everything should look pretty and, as already mentioned in section 4.2.2, the actions should be straightforward and visible. Implementing a nice style can require a lot of code. The code for these styles can become a big part of the implementation. Hence, a good framework is necessary to reduce the lines of style code to a reasonable number and to help to improve the readability of the code.

[7] `https://github.com/d3/d3-hierarchy`

Tailwind CSS[8] is a framework that assists developers to style their application. The difference with more famous frameworks, like Bootstrap, is that Tailwind CSS has no default theme. If you want to use a Bootstrap-feature, this eventually comes along with other features you do not always want and it can be quite hard to undo the part you do not want. Furthermore, you have to write additional lines of style code to undo the unwanted parts. With tailwind on the other hand, you can grab only the features you want, without side-effects. Figure 5.1 shows an example with two small listings. The first uses inline style while the second makes use of tailwind CSS.

```
<div
  style={{
    position: absolute,
    border: 1px solid black,
    borderRadius: 0.25rem,
    padding: 0.5rem,
  }}
/>
```

Listing 5.1: Normal CSS, no tailwind.

```
<div
  className={
    'absolute border
    ↪ border-solid
    ↪ border-black rounded
    ↪ p-2'
  }
/>
```

Listing 5.2: With tailwind CSS.

Figure 5.1: Difference when using tailwind CSS or not.

The figure illustrates the difference to implement four CSS property-value pairs in normal CSS and implementing the same four with tailwind. In the case of normal CSS, we need four lines of code to obtain the intended result. On the other hand, with tailwind CSS, we add some classes providing the same style. By using these classnames, the developer can see the properties faster than with normal CSS. Hence, in case of lots of lines of CSS code, tailwind can be faster to use compared to normal CSS.

Bootstrap can be very interesting to use in applications and websites that should run on devices with small screens (e.g. smartphones). But we decided not to tailor our application for such devices (section 4.2.3). Given the example and the fact that Tailwind CSS does not have a default theme, we prefer Tailwind over Bootstrap. Keep in mind that it is certainly possible to create the same application with Bootstrap, but with Tailwind CSS, the code will be easier to understand.

5.3 Main Code Structure

With the technologies mentioned in the previous section, the most important pillars the application relies on are discussed. In this section, we will explain the main structure of the code, such that it is clear how all elements work together. Figure 5.2 shows a visualization of the structure of the code.

[8] https://tailwindcss.com/

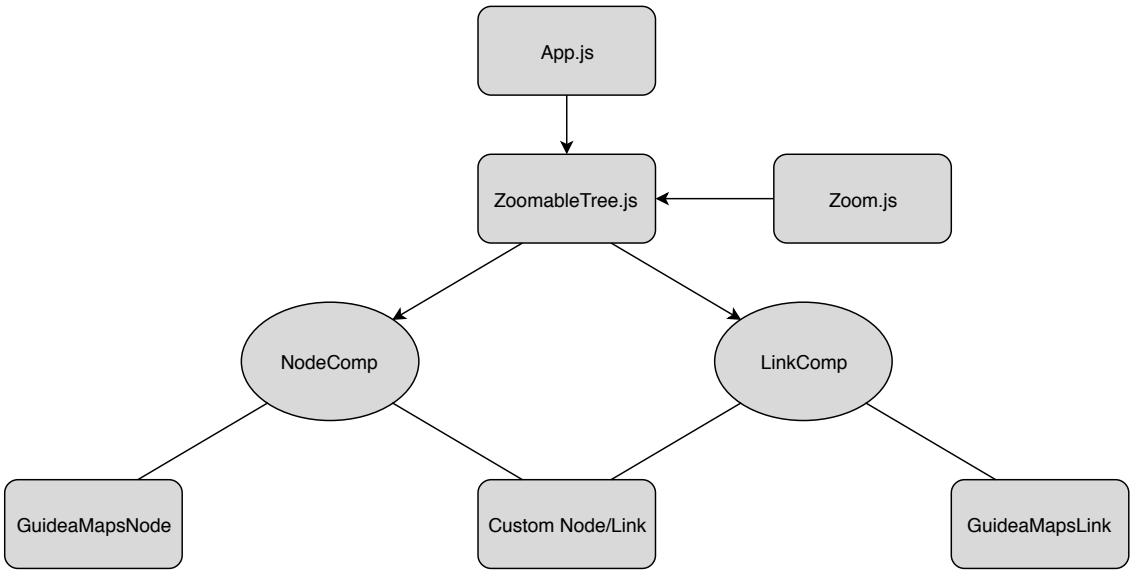


Figure 5.2: Structure of the application.

The application is developed in such a way that it can be used as a library for other purposes than GuideaMaps. In this way, Research Goal 4 to have a generic solution will be achieved. The general, the always-returning part of the code can be found in App.js, ZoomableTree.js and Zoom.js, where the layout of the nodes is defined as well as the implementation to allow the user to zoom the visualization in and out. To realize the library, the layer of the NodeComp and LinkComp is very important. For GuideaMaps (shorthand GM), an implementation for NodeComp and LinkComp, being GMNode and GMLink, is provided. Each implementation describes how every node and every link should look like in the visualization. When a particular user would like to have a different representation for the nodes or the links or both, new components (e.g. MyCustomNode and MyCustomLink) should be implemented. In the rest of the code, only one line should be adapted: in App.js, ZoomableTree is called with a certain number of props. Two of these props are NodeComp and LinkComp, which are set to GMNode and GMLink, respectively, by default. Hence, the only action that is required to *plug in* an other component is replacing GMNode and GMLink by this component (e.g. MyCustomNode and/or MyCustomLink). Hence, the visualization can be customized to the needs of the user and FR 3 is achieved. Figure 5.3 shows the part of the code in App.js that should be adapted as explained. Note that the shown props are not the only props that are passed to ZoomableTree. The others are omitted for readability.

The reason why we chose for this approach is that now the developer does not have to change anything of the default implementation. He only has to create his own components and plug them in as props for ZoomableTree and leave the rest of the implementation as it is.

```
<ZoomableTree
    NodeComp={GMNode}
    LinkComp={GMLink}
/>
```

Listing 5.3: Default components.

```
<ZoomableTree
    NodeComp={MyCustomNode}
    LinkComp={MyCustomLink}
/>
```

Listing 5.4: Custom components.

Figure 5.3: Two listings showing how to use the library.

Next to custom components, a developer can also implement his own functions to handle changes in the visualization. For example, to add a new child node, GuideaMaps calls the function passed to the *onAddNode*-prop (i.e. addGMChildNode). In a custom implementation, you can pass another function to this prop to make sure that other work is done. If you do not want to allow users to add child nodes, you do not have to remove this line but you just replace addGMChildNode by *() -> null*, a function that does not do anything. It is not wrong to keep addGMChildNode as parameter, but if the function would be called, this can lead to errors or wrong behaviour of the application. Further, the library is created in such a way that the default implementation should not be changed. Hence, developers should not change the implementation of functions like addGMChildNode. Therefore, we recommend to always pass *() -> null* as argument to functions that should not be used and to implement custom functions if other behaviour is needed. More details about the use of the library can be found in chapter 6, where a complete use case is elaborated.

5.4 Default Implementation: GuideaMaps

Now the overview of the structure of the application has been discussed, we will consider the GuideaMaps visualization and its implementation details in this section. An example visualization can be seen in Figure 5.4.

The nodes represent a specific part of the data and the links illustrate the relation between the data (i.e. the nodes). Before we discuss the nodes into detail, we start with the *navigation bar* above the visualization itself. This navigation bar is divided in three equal parts. In the center, the user can select via an option menu which map or template he wants to use. The name of the selected option is always visible. This functionality is available because the user should be able to switch between different tasks. In each task, the user can work with a different map or start a new map based on one of the existing templates, created by a map creator.

The right part of the navigation bar contains a single button. A click on this button makes sure the visualization is zoomed in such a way that all nodes fit in the window. This is one of the reasons why we mentioned in requirement OR 1 that the screen of the used device should not be too small. If the screen is too small, the nodes of a larger visualization will overlap each other because otherwise they would

not fit all on the screen. When nodes overlap, a lot of information can be hidden and the visualization can become useless.

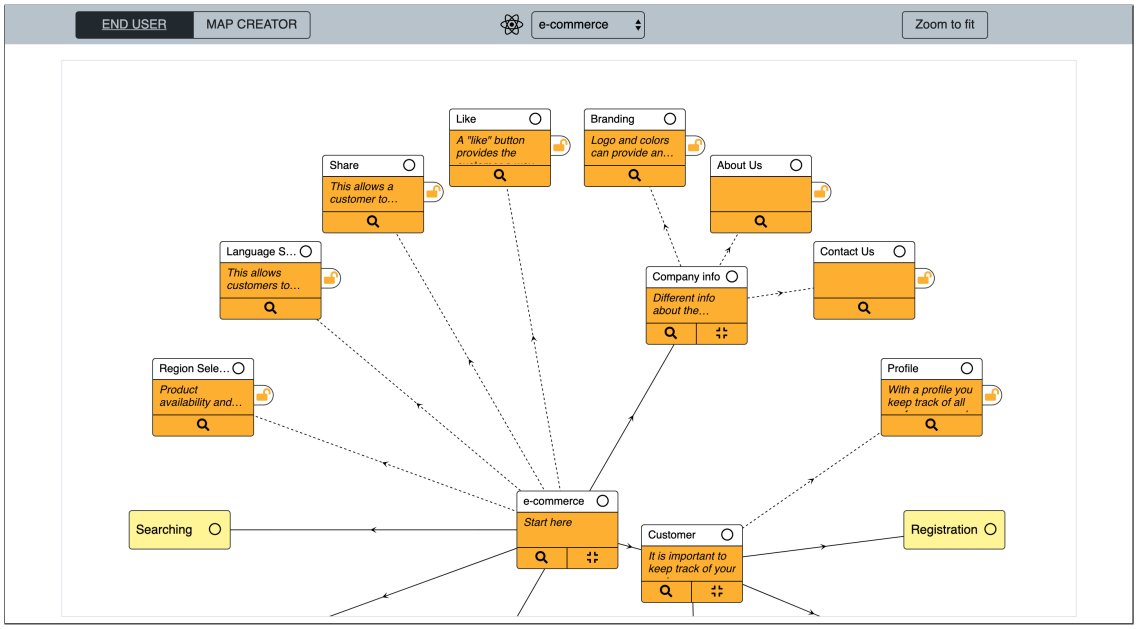


Figure 5.4: GuideaMaps Layout.

The left part of the navigation bar is created to be able to switch between the user modes (i.e. end-user or map creator). As required, a map creator has more rights and hence he can perform more actions than an end-user. In the following subsections, the differences between these modes are elaborated.

5.4.1 Regular Nodes

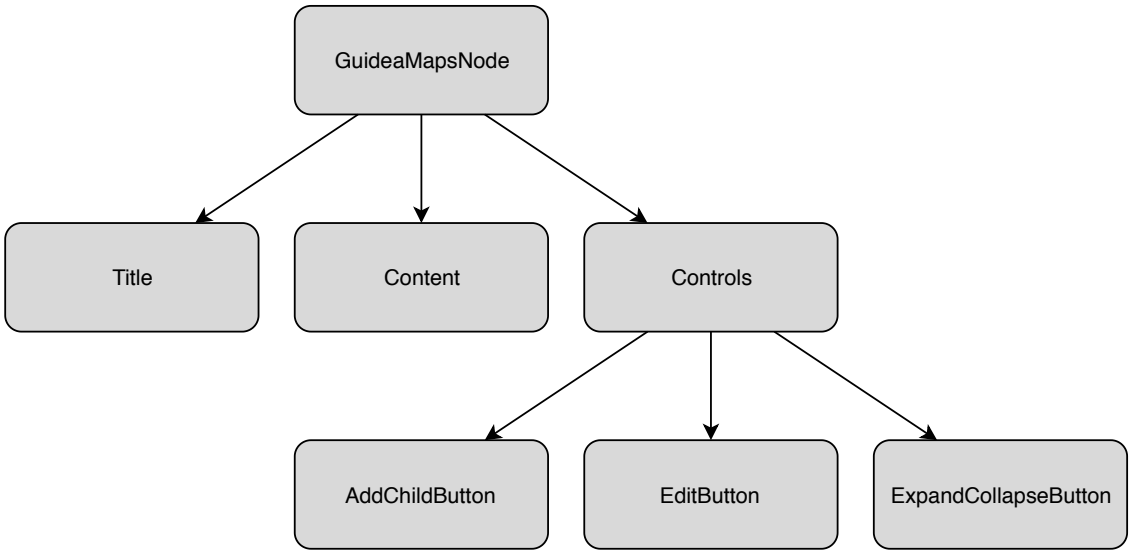


Figure 5.5: Structure of GuideaMapsNode.

As you can see in Figure 5.5, the structure of a GuideaMapsNode node is quite simple. Each node consists of three html *div*-elements: a title-div, a content-div and a controls-div explained below.

5.4.1.1 Title

The title-div is positioned at the top of the node. It consists of the title text when this is given by the map creator. Otherwise, it shows *Insert title* in italics to remind the user that he still has to set a title for this node. An end-user can set the title if no title exists yet, but he can never change it. The reason for this is because we distinguish two situations. In the first situation, the end-user adds a child node for a particular parent. This node has no title by default and then he should be able to insert a title. In the second situation, the user opens an existing node created by the map creator. In that case the end-user is not allowed to change the title of the node, because otherwise he would be able to change the meaning of the node.

Next to the title-text, the title-div contains a so-called "completeness-icon" placed near the right border. This icon indicates to an end-user whether all information for the node, i.e. content, is provided or not. The icon that is shown also depends on whether the information in the child-nodes is given or not. We distinguish three possible situations:

1. The node itself and all of its (non-optional) children are correctly filled in. In this case the icon will be a completely filled circle.

2. The node itself and all of its (non-optional) children are still empty. In this case the icon will be an empty circle.

3. In all other cases, the node itself or at least one of its child-nodes is filled. Then the icon will be a semi-filled circle.

This icon can assist the end-user in determining whether all information requested by a template is given. For example, if he checks the root node and sees that the circle is completely filled, he knows that all mandatory information in every child-node is given. On the other hand, suppose only one node does not yet contain the required content, then the user starts from the root node and always follows the child-node with a semi-filled circle. In this way, he will find the incomplete node much faster than in the case he has to check all the nodes, one by one. Hence, the icon is an element that helps to improve the usability of the application.

5.4.1.2 Content

The content of each node is some text describing the information required for the node. As long as no content is provided by the end-user, the description provided by the map creator is shown in italics to instruct the end-user which content to provide.

5.4.1.3 Controls

The last part of a node is the controls-div. With *controls* we mean the different actions a user can take concerning the particular node. As actions, a user can (a) add child nodes, (b) explore and edit data of a node, (c) expand and (d) collapse a node. As mentioned in requirement UR 2, it is very important to choose for unmistakable icons. Therefore, we chose for the following icons (Figure 5.6).

(a) Plus (b) Explore/Edit (c) Expand (d) Collapse

Figure 5.6: Good icons for the following actions: (a) add child node, (b) explore and edit node, (c) expand node and (d) collapse node.

Users can perform three types of actions. Each action can be performed by a click on its consecutive button. Hence, there is a button to "open" the node to view and edit the data, and one to expand or collapse the node, i.e. to show or hide the child nodes, respectively. When a node is collapsed, all child nodes on all lower levels in the hierarchy are hidden. On the other hand, when a node is expanded, only the child-nodes of the next level in the hierarchy are shown.

The third button that can be found in the controls-div of the node is to add a child node. A click on this button opens a small modal window, where some information about the new node should be provided before the node can be created. Figure 5.7 illustrates what that modal window looks like when the user wants to add a regular node. He has to start by selecting the option "Regular" at the top, after which two input fields appear to insert the title and the description for the node. A click on the button at the bottom will add a child node of the selected type and with the provided title and description.

It is possible to add an *empty* node if the user does not provide a title and/or a description. Even though it is not really useful to add nodes without any context, we only require the user to select the type of the child node. If the type is unknown, the node will not be added. A situation where a map creator would add an empty node is when he wants to remind himself that a child node is necessary at that place, but details about the expected content are not yet known.

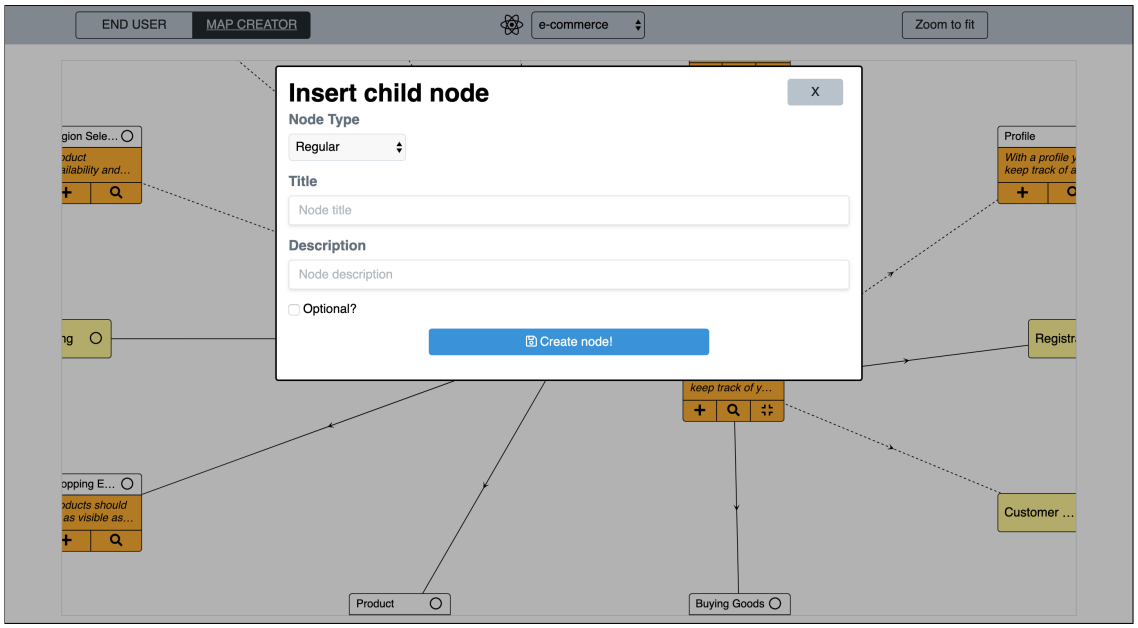


Figure 5.7: Adding a regular node.

5.4.1.4 EditModal Window

A click on the button to open the node opens a modal window containing the data. What this modal window looks like depends on the mode. If we are in map creator mode, a form will be shown to allow to change the title, the description, and the background color (Figure 5.8). In the case of the end-user mode (Figure 5.9), only the content and the background color can be changed. Only the first time a child node is added and no title and description is available yet, the end-user is able to fill in these data as well.

When changing the background color, the user can choose to use the color also for all children or not. If the checkbox "Include children" is checked, the background color of all child-nodes on all sublevels in the hierarchy will be changed to the new color. Otherwise, only the background color of the current node will be changed.

The text color is black by default. This can become a problem when the background color is changed to a dark color and certainly when it is black as well because then the text is not readable anymore. To solve this issue, we wrote some CSS rules that change the text-color depending on the background color. A black (or dark) background will result in white text and a white (or light) background results in black text.

The possibility to change the background color of the nodes and the fact that the text color is adapted taking the background color into account is also a useful feature for colorblind people. In this way, they can adapt the background colors of the nodes to colors they can distinguish better. Hence, requirement UR 4 is achieved by this functionality.

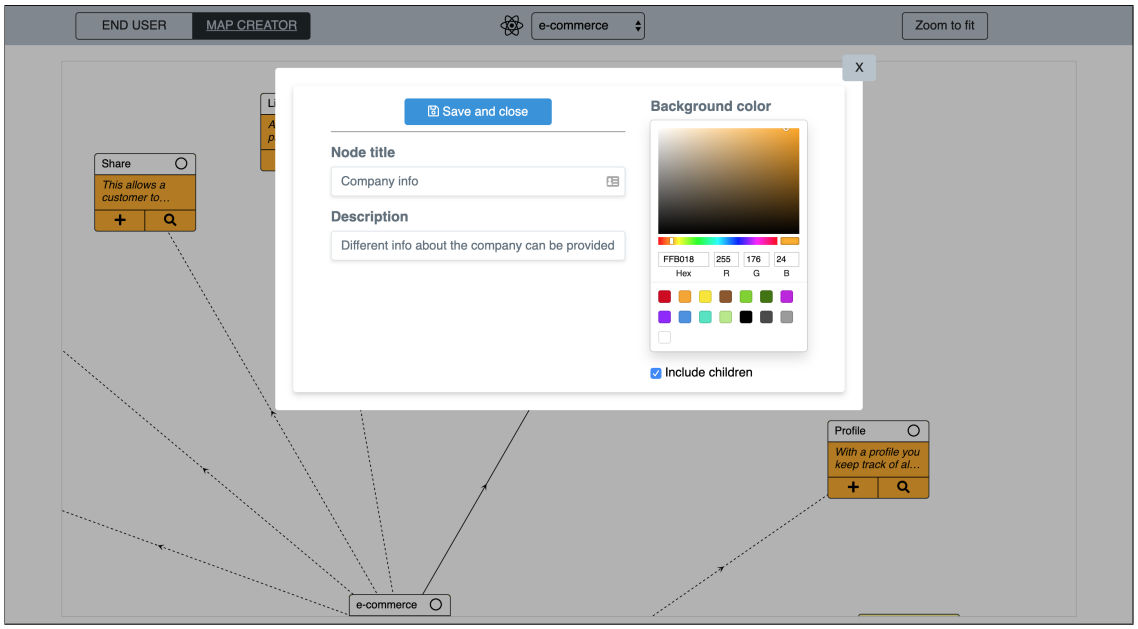


Figure 5.8: The edit modal in map creator mode.

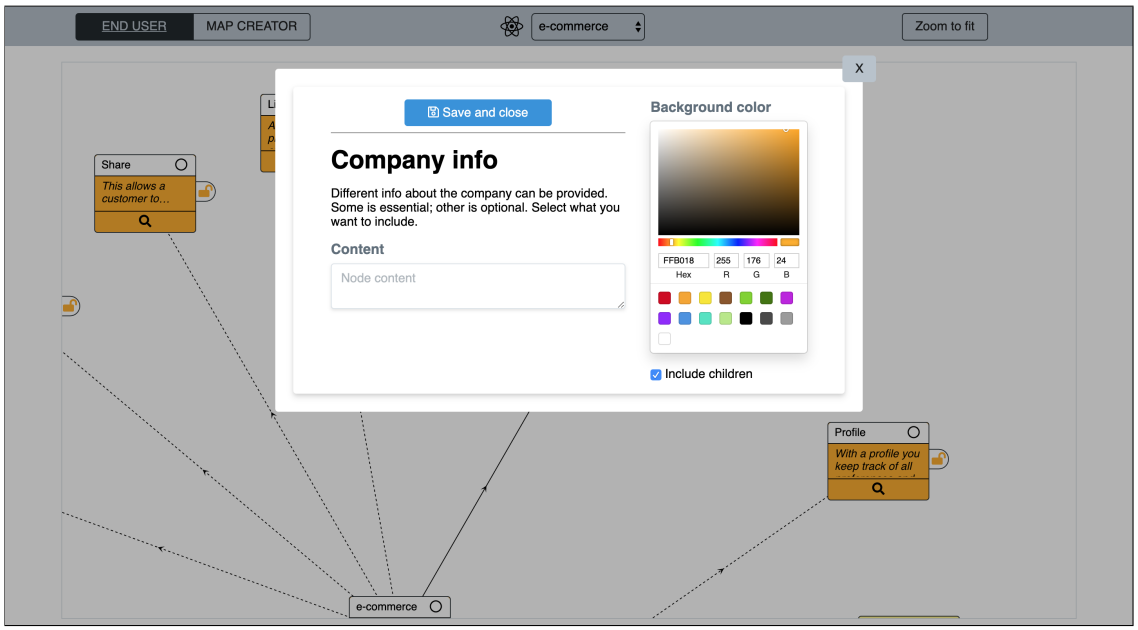


Figure 5.9: The edit modal in end-user mode.

5.4.2 Choice Nodes

As explained in section 2.1.1, choice nodes differ from the regular nodes in multiple ways. First the layout is different: while regular nodes have a title-, content- and controls-div, a choice node only has a title-div.

Remember that the purpose of a choice node is to let the user choose between different possibilities as children for the node. For example, the choice node with

title “Payment Method” can have “Cash”, “Bancontact” or other methods as child node. Depending on the situation, one (or more) of these methods can be chosen by the end-user.

It is the task of the map creator to define the possible choices the end-user can select. Therefore, when the map creator clicks on a button to add a child node, a similar modal window opens as shown in Figure 5.10. The modal window provides text fields in which the map creator can insert a title and a description for the choice node, as well as the information for each choice, i.e. a title and a description. Because there is no maximum limit on the number of choices a map creator can provide, he can always ask to add more text fields to be able to give the information about an additional choice.

Another feature particular for choice nodes consists of the cardinalities a map creator can define. The map creator can set a lower limit and an upper limit. The lower limit is the minimum number of choices the end-user has to select. For example, if the lower limit is equal to two, the end-user cannot select only one choice. The upper limit is the maximum number of choices that can be selected by the end-user. By default the lower limit is equal to zero and the upper limit is equal to the total number of choices the map creator provided. It is not possible to set the upper limit equal to zero because this would not make sense: end-users would not be able to select any choice. Hence, the choice node would have no meaning and be useless.

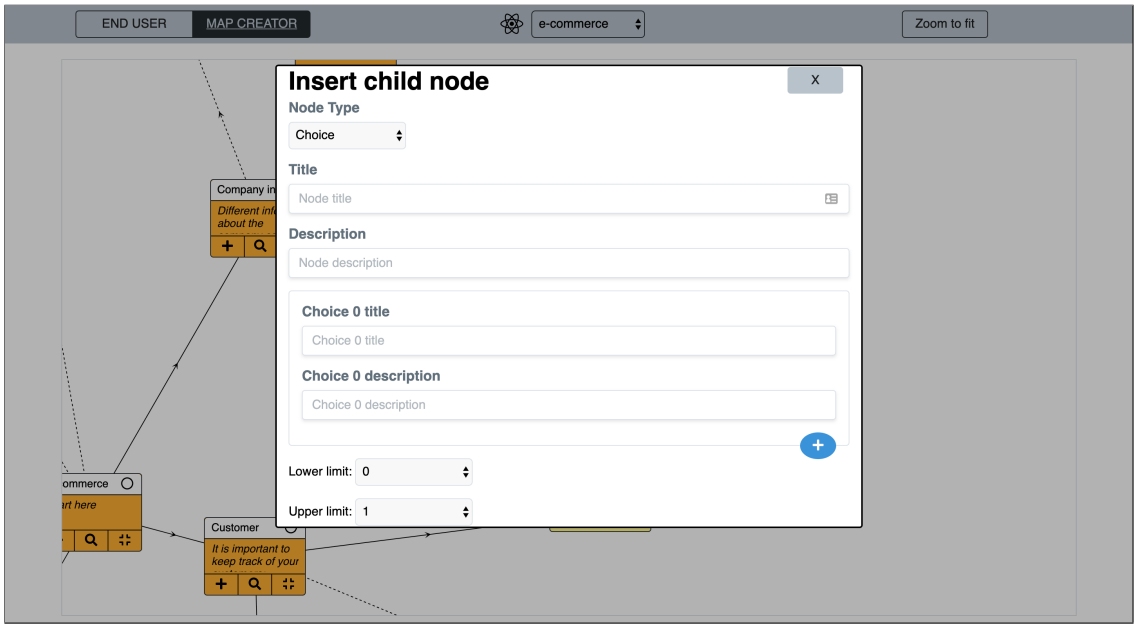


Figure 5.10: Adding a choice node.

After the node is created, the map creator can edit the node by performing a single click on the choice node. Then a modal window opens and he can edit the title, the description and the already defined choices (Figure 5.11). To delete a choice, the map creator only has to empty the text fields of the choice.

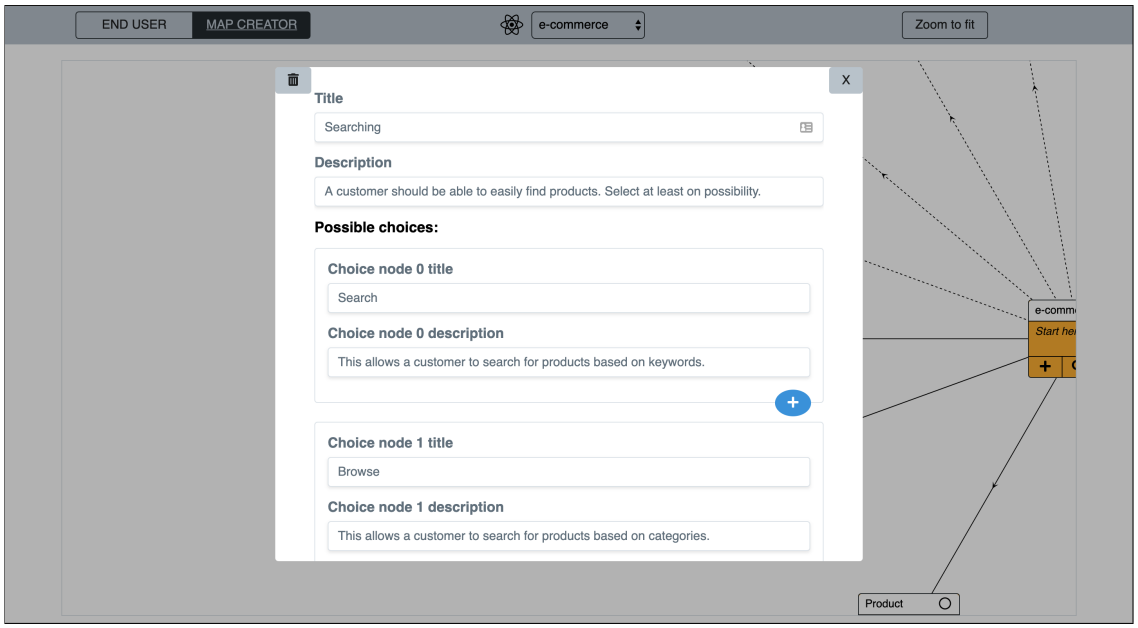


Figure 5.11: An opened choice node in map creator mode.

The modal window showing up when the end-user clicks on a choice node (Figure 5.12) is not equal to the modal window the map creator can see after the same action. The end-user sees the title and description of the node on top, but (s)he cannot edit these like a map creator. Under the description is shown how many choices the end-user should select. For example "Select between 1 and 3 choices", where 1 is the lower limit and 3 the upper limit set by the map creator. This instruction is followed by a form, where every possible choice is made visible by showing the title and the description. The end-user can select the choices he wants via a checkbox. With a click on the button, the selected choices are registered and added as children to the choice node. It is possible at any time to adapt the choices by reopening the modal window. At that moment, the checkbox(es) of the previously selected choice(s) will be checked. To remove a choice as child of the choice node, it is sufficient to uncheck the checkbox.

Next to selecting choices given by the map creator, it can sometimes be necessary to be able to add an extra possibility, which was not given by the map creator. Therefore, an end-user can create additional choices, which he can select later on (the upper limit is incremented for each added choice). Additional choices can be created in the same way as the map creator can do it. By a click on the button, an additional part of the form appears in the modal window. When the form is submitted, the additional choice is registered with the given title and description. An example is shown in Figure 5.13.

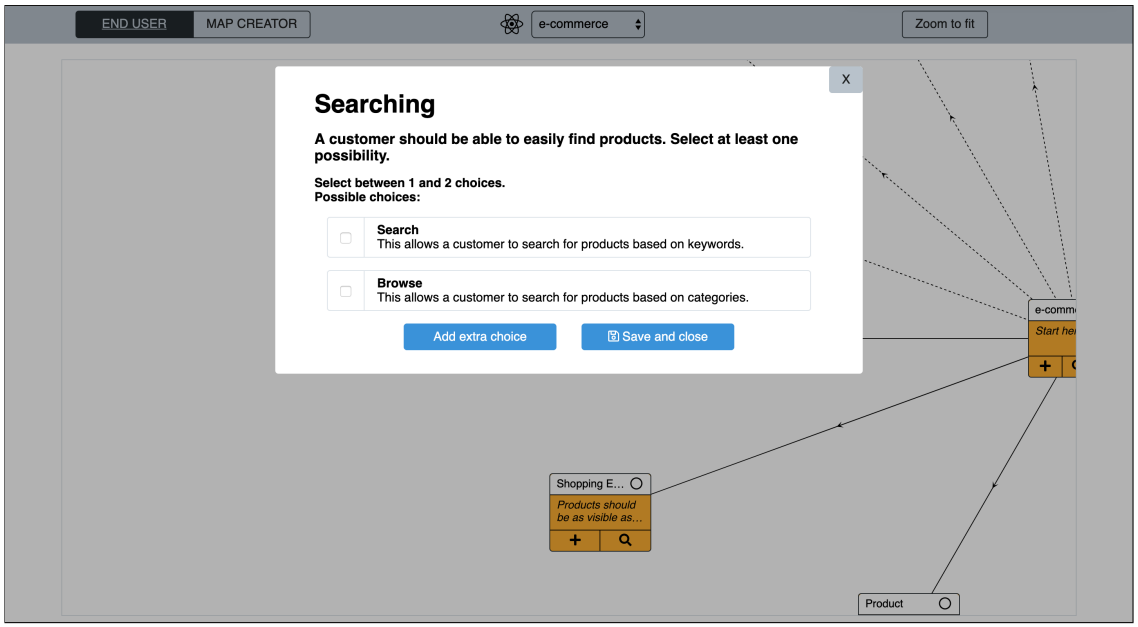


Figure 5.12: An opened choice node in end-user mode.

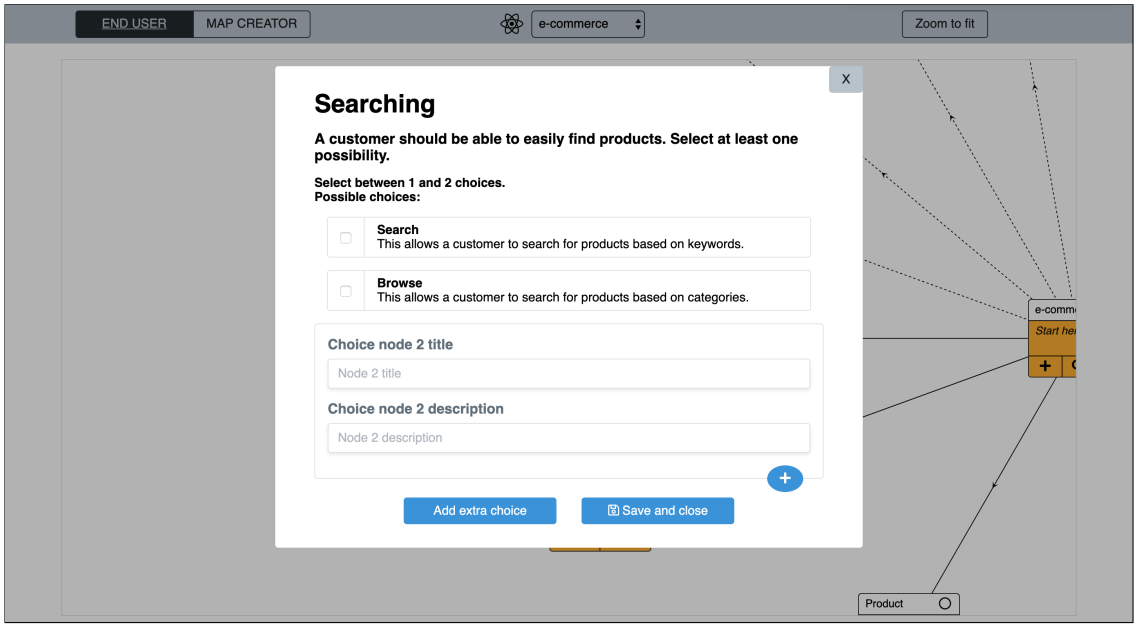


Figure 5.13: An opened choice node in end-user mode with possibility to create an additional choice.

5.4.3 Optional nodes

A map can also provide optional nodes. They can be recognized by the dotted link between this node and its parent. A special characteristic of optional nodes is that these nodes can be disabled by the end-user. Therefore, optional nodes contain an additional button, represented as a lock, at their right side. As long as the node is enabled, the button is represented by an open lock. A click on this button disables

the node and changes the icon into a closed lock. The node can be enabled again by clicking the lock again. Next to the lock icon, a disabled node can be recognized by its gray background and blurred text so that the content is less readable.

A second characteristic of optional nodes is that they do not contribute to the "completeness icon" of the visualization. Hence, to determine whether the completeness icon of a node should be empty, filled or partially filled, optional child nodes and their descendants (which are optional as well) are skipped because it does not matter whether content is provided in these nodes or not. There is only one exception on this rule: when a choice node is optional but it does have children (selected by the end-user), these children are not optional and are checked for completeness.

6 Another Use Case

To demonstrate that we achieved Research Goal 4, we implemented a different use case for the system called *Plateforme DD (PDD)*. In the first section of this chapter, we describe the purpose of this use case and explain Plateforme DD. The second section is about how the library could be used to create a visualization for this system.

6.1 Plateforme DD

PDD is a ULB (Université Libre de Bruxelles) project with a scientific mission, i.e. durable development in Brussels. On the one hand, the project tries to arouse interest in sciences among pupils in Brussels. On the other hand, it wants to make it easier (for everyone, not only pupils) to understand the situation in Brussels and to make them take efforts for a durable development in the city.

The different pages on the website[1] of Plateforme DD correspond to projects the pupils worked on in small groups. The content on every page can be seen as a small report concerning what they learned about the topic. Many of these topics are linked to each other in some way. However, it is absolutely not easy to understand the underlying tree structure by browsing through the website. On each page corresponding to one of the topics, a number of links can be found. But by following these links, the user can get lost quite easily on this site and it is very difficult to understand how the different projects are linked to each other. Hence, the project owners wanted to have a better visualization of the relationships between the different Web pages. Therefore, we visualized the tree structure by creating a separate node for each (sub)topic on the site. In this way, it should be easier for

[1] https://sciences.brussels/dd/

the user to grasp the relationships between the different projects. To create such a visualization, we could make use of our library. In this way, we can show that our tool can be used in cases where the visualization and functionality is completely different from the one for GuideaMaps. The created visualization for the content of the Plateforme DD website can be seen in Figure 6.1.

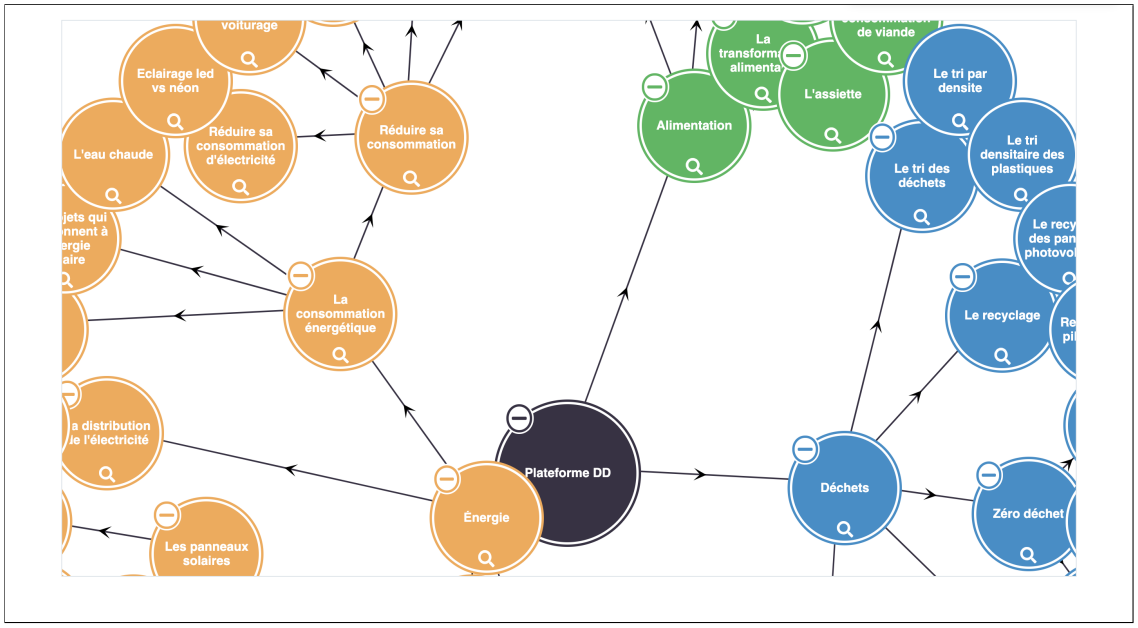


Figure 6.1: Plateforme DD Visualization.

6.2 Customization Effort

In Figure 5.3, we showed how we could alter the visualization of the data without affecting the default code. Remember that the props in that figure are not the only customizable props of the library. Figure 6.2 shows the differences in configuration when we compare the visualization of GuideaMaps with the one of Plateforme DD.

```
<ZoomableTree
    NodeComp={GMNode}
    LinkComp={GMLink}
    EditModalComp={
    ↪ GMEditModal}
    onAddNode={addGMNode}
    onDeleteNode={
    ↪ deleteGMNode}
    onNodeUpdate={args ->
    ↪ updateGMNode(args)}
    onVisibleChildrenUpdate={
    ↪ nodeId =>
    ↪ updateGMVisibleChildren(
    ↪ nodeId)}
/>
```

Listing 6.1: GM Configuration.

```
<ZoomableTree
    NodeComp={PDDNode}
    LinkComp={PDDLink}
    EditModalComp={
    ↪ PDDEditModal}
    onAddNode={() -> null}
    onDeleteNode={
    ↪ deleteGMNode}
    onNodeUpdate={args ->
    ↪ updateGMNode(args)}
    onVisibleChildrenUpdate={
    ↪ nodeId =>
    ↪ updateGMVisibleChildren(
    ↪ nodeId)}
/>
```

Listing 6.2: PDD Configuration.

Figure 6.2: The configuration differences between GuideaMaps and Plateforme DD.

6.2.1 NodeComp

The most obvious difference, which can immediately be seen when comparing both visualizations, is the layout of the nodes. In GuideaMaps (GM), we had rectangular nodes, while in Plateforme DD (PDD) the nodes are circular. To achieve this result, we implemented a special component, called *PDDNode*, to replace *GMNode*. From then on, every node in the data will be visualized by the code of PDDNode instead of the code of GMNode. Hence, GMNode still exists; it is not deleted or overwritten, but simply not used in this configuration.

6.2.2 LinkComp

The second prop is responsible for the representation of the links. You will not observe big differences between a GMLink and a PDDLink, except for the thickness: a PDDLink is thicker than a GMNode. As a consequence, the implementation of PDDLink is a copy of the GMLink with the value for *strokeWidth* as only difference.

If the user does not need a different style for the links, he does not have to create a PDDLink component. In that case, it is possible to use GMLink as LinkComp, while the other props are eventually specially created for Plateforme DD.

6.2.3 EditModalComp

The edit modal of Plateforme DD is completely different compared to the one of GuideaMaps. First of all, it is not a real *edit*-modal because a user of the PDD

visualization is not allowed to update the data of the nodes, he can only consult the available data, represented as in Figure 6.3.

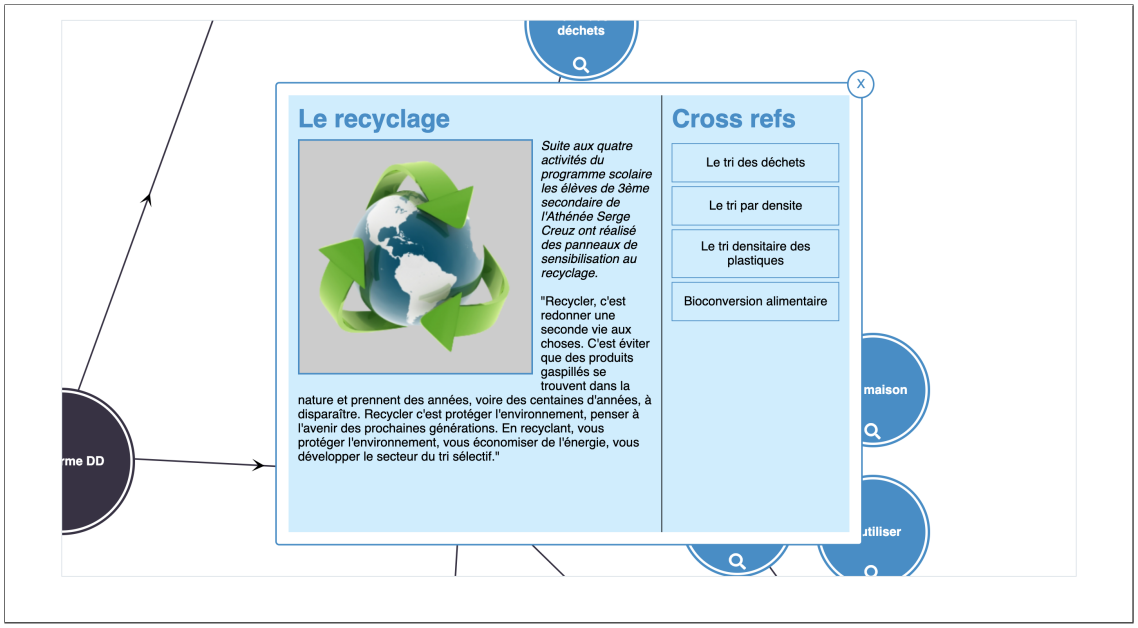


Figure 6.3: Plateforme DD Edit Modal.

At the left part of the modal, the actual content is shown. It starts with the title of the node, followed by a picture. If no picture exists, the text providing more information about the topic of the node is shown immediately under the title. Otherwise the title and the text are separated by the picture.

The size of the modal is fixed, but in case the information does not fit the modal, the user can scroll in the left part to continue reading until the end. The modal will never grow in size to make sure all data fits in it.

For some nodes, the modal will contain a list of so-called "cross references", which are links to nodes that in some way are related and which are not a direct child of the node. Such a link is not shown in the visualization, but by providing such a list, the user can still discover them. A click on a cross reference will close the modal and *move* the visualization until the node of the corresponding cross reference is centered. This allows the user to navigate to linked nodes without the need to show a spaghetti of links in the visualization.

6.2.4 Action Listeners

In GuideaMaps, we have a controls-div (section 5.4.1.3) containing a number of buttons to allow the user to perform some actions. In Plateforme DD, we chose for another approach. Because we have circular nodes instead of rectangular, it is less convenient to position multiple buttons next to each other at the bottom of the node. Further, we don't need a button to add child nodes because this is not allowed in Plateforme DD. Hence, we only have two buttons: one to expand and

collapse the child nodes and one to open the edit modal.

The button to open the node is positioned in the center at the bottom of the node. A click on that button opens the modal and centers the node such that it is in the middle of the screen when the user closes the modal. Specific to this implementation is that there is a second way to open the modal. If the user clicks once on the node, it is centered. If he clicks once more on the node (not necessarily on the button), the modal will open and the content will be shown. This is because a click on a centered node results in the same action as a click on the button to open the node.

The second button is one to expand or collapse the child nodes. As already said, we cannot position it next to the other button at the bottom of the node. Therefore we position it at the top-left of the boundary of the node. There, a circle is visible with a minus inside if the children are visible. A click on the minus will collapse the node such that the children are not visible anymore and the minus will be replaced by a plus. When the user clicks on that plus again, the exact opposite action will happen: the node will be expanded (i.e. the children of the node will be visible again) and the plus will be replaced by a minus.

Note that a plus and minus can be used here as symbols to expand and collapse a node. This is possible because there can be no confusion with the action of adding a child node in Plateforme DD. If this would be possible, other symbols should be used to avoid such confusion and to not reduce the intuitiveness (requirement UR 2).

7 Evaluation

For our tool, usability and user experience are very important. Therefore, we did two user studies. The first was to test the usability of GuideaMaps 2.0 (as map creator and as end-user), while the second study more specifically evaluated whether the visualization provided for Plateforme DD was an improvement over the original structure provided on the website.

This chapter is divided in two sections. The first describes the user study of GuideaMaps 2.0 together with its results. In the second section, the same is done for the user study of Plateforme DD.

7.1 GuideaMaps

In this section, we describe the evaluation of the default implementation of the tool. With "default" we mean that no implementation is plugged into the library but the standard visualization is used as described in section 5.4.

7.1.1 Setup

The evaluation was split in two parts, each evaluating one of the two modes. In the first part, we evaluated the map creator mode with eight participants. Because GuideaMaps is intended for the requirements elicitation of domain-specific software, the map creators will in general be people from the domain of Computer Science. Therefore, we asked people with a background in Computer Science and/or visualization techniques to participate. However, the users that will provide the actual requirements can be almost everyone. Therefore, a Computer Science background was not a requirement to participate in the part evaluating the end-user

mode, which was evaluated by eight participants. All participants for the evaluation of both modes were unique, i.e. there was no participant who evaluated both modes.

All participants answered the questions at home without additional explanation than the text provided at the beginning of the questionnaire.

7.1.2 Tasks & Questions

For the two modes, a different (online) questionnaire was created (see Appendix A and Appendix B). Both questionnaires start with a brief explanation about the tool.

Next, the participant was asked for his age and whether he had a background in Computer Science and/or visualization techniques or not. These general questions were followed by a number of tasks the participants had to perform followed by some questions.

The goal was to let the participants experience the actions a real map creator or end-user can perform. After each action, we asked to indicate how intuitive it was for them to obtain the requested result. We asked this question because we want the tool to be usable without much explanation or training. When the majority of the participants would indicate that a certain action was not intuitive at all, we know something should be changed in the design of the tool. After all these tasks were finished, the participant was asked to upload a screenshot of his screen, so that we can check whether everything went as intended.

Furthermore, to obtain a general idea of the usability of the tool, the participants were afterwards asked to fill out the SUS usability questionnaire[1]. We chose for this questionnaire because, according to a study of Tullis & Stetson (2004), SUS gives the most reliable results in comparison to other questionnaires (e.g. QUIS and CSUQ). Further, SUS is the only questionnaire of the ones they studied that addresses multiple aspects of the user's reaction to the system (in their case a website), while other questionnaires ask more about specific features of the system.

The questionnaire was concluded with some open questions. We asked to tell us what was good about the visualization and where they think we could have done better.

7.1.3 Results

Because there are different questionnaires for the two modes, the results are separated in the next two sections. For the questions about the intuitiveness of the task, we only discuss remarkable results or questions of which the result was (absolutely) not intuitive.

[1] `https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html`

7.1.3.1 Map Creator

The first task to evaluate the functionality of the map creator was to provide a title to the root node. We see that 25% of the participants indicated this action was not intuitive and another 25% were neutral about the intuitiveness of this task. A frequently returning remark was that participants tried to click on the title to edit it. They also said they did not link the magnification icon to editing possibilities. Suggestions for better icons were a gear or a pencil. Hence, we should consider to replace the magnifier icon with a better icon.

The other tasks were evaluated as intuitive to very intuitive, with some exceptions indicating *neutral*. However, the task to delete an option of a choice node was not intuitive for 87,5% and absolutely not intuitive for 12,5% of the participants. This means we could have done much better for this functionality. A comment we encountered frequently was that a small button to delete an option would be much more intuitive than having to delete the title and description. Now, the functionality is somewhat hidden and difficult to find. Another remark about buttons is that the position of the button to delete a node was weird.

Finally, the participants answered the ten SUS questions and the results are satisfying: the average score is 72,5 on 100.

7.1.3.2 End User

As already mentioned, different tasks and questions were prepared to evaluate the functionality provided to the end-user. In the first task, it was asked to add content to a specific node. While 75% experienced this as a (very) intuitive action, 25% found it not intuitive. This has possibly a similar reason than in case of the map creator: the participants probably clicked on the place of the node where the content should be visible, but could not insert the content from there.

Further, when some options of a choice node are selected, child nodes are added to the visualization. We asked the participants to delete one of these options. While 5% indicated this action is (very) intuitive, 25% experienced the same action as not intuitive and 25% as neutral. Hence, deleting an option was not super clear for half of the participants.

We also asked whether the participants understood the meaning of the "completeness-icon" (without calling it like this to not reveal the answer). Some of the answers were quite close to the correct interpretation but no participant could give a completely correct explanation. The same situation took place for the question about the lock symbol. To solve this issue, some participants suggested to provide a document with guidelines explaining possible actions and the meaning of elements like the completeness-icon.

In the end, the participants answered the ten SUS questions and with an average score of 69,375 on 100, the result is slightly lower than the one of the map creator.

7.1.3.3 Overall

In general, the user studies yielded important information. First of all, we have to be even more careful than we expected with the icons we use for actions (UR 2). We saw that users do not click on a button if they are convinced it is not meant for what they are searching for. Even though some participants mentioned the tooltips were helpful, some of them probably did not notice them and misunderstood the meaning of the icons.

In the future, deleting elements should always be done via a button. Other ways are not intuitive (enough). Further, a frequent remark was that participants tried to click on the title or the content of a node to change it. Hence, it would be useful to be able to change it in this way instead of having to open the modal for every change.

As a conclusion, we can say that our application scores reasonably well in terms of usability. However, some issues need to be solved to improve the tool (e.g. icons, guidelines).

7.2 Plateforme DD

The goal of the evaluation of the Plateforme DD use case (chapter 6) was different in comparison to the one about GuideaMaps. Here the goal was to investigate whether our visualization would be easier to use and understand than the current website[2] of Plateforme DD for the target audience, being pupils from secondary schools.

In this evaluation, the participants (52 in total) were around 15 years old. This age group was chosen because the content on the current website was created by youngsters of similar ages. Note that the participants of this user study never worked with the website before.

7.2.1 Setup

We divided the participants into an experimental and a control group. The control group solved the questionnaire using the website, while the experimental group solved the same questionnaire using our application.

There was one tablet per two children on which the website or the application was running (depending on the group they were in). Note that they did not work with both systems to avoid a learning effect (having used one system could influence the results for the other system).

[2] `https://sciences.brussels/dd/`

7.2.2 Tasks & Questions

The user study consisted of different questions the children had to answer for which they had to explore or search in the system. Before they started with the actual tasks, we gave them five minutes to explore the system, i.e. the website or the application, so that they could become somewhat familiar with the system. We mentioned that it was important to understand the structure as much as possible because some questions would follow after this introduction period.

After five to ten minutes, we distributed the papers with the tasks and the questions. With other words, all children (of both groups) received part 1 of the questionnaire (see Appendix C). The first six questions of part 1 intended to test the understanding of the structure of the content. Then, four more questions were added to test the ease of searching and finding information in the system. The participants using our application also had to answer four more questions to measure their understanding of the visualization approach used.

After twenty minutes, all questionnaires of part 1 were collected, even if the participant did not finish all questions yet. Together with the questionnaires, the tablets were collected, as they were not needed anymore for part 2 of the user study. After collecting everything, part two of the questionnaire was distributed to all participants. In this part, the questions were not about the content or the structure of the information, but about usability and user experience. We did this because we also wanted to be able to compare the results about usability and user experience of the two systems. The questions about usability and user experience can be found in Appendix C and Appendix D, respectively. The usability questions are based on the SUS questionnaire previously mentioned, but adapted a bit because not all questions were relevant for our purpose. To evaluate the user experience, we used the UEQ questionnaire[3].

7.2.3 Results

In general, we hypothesize that participants using our application answer more of part 1 correctly than the participants using the website.

Note that the participants only got 20 minutes to answer all questions. We expected that not all participants using the website would be able to answer all questions in this amount of time. Even though we know it is possible that some participants using our application would also not be able to finish the questionnaire either, we predict that participants using the application will be able to answer more of the questions than participants using the website.

[3] `https://www.ueq-online.org/`

7.2.3.1 Structure Comprehension

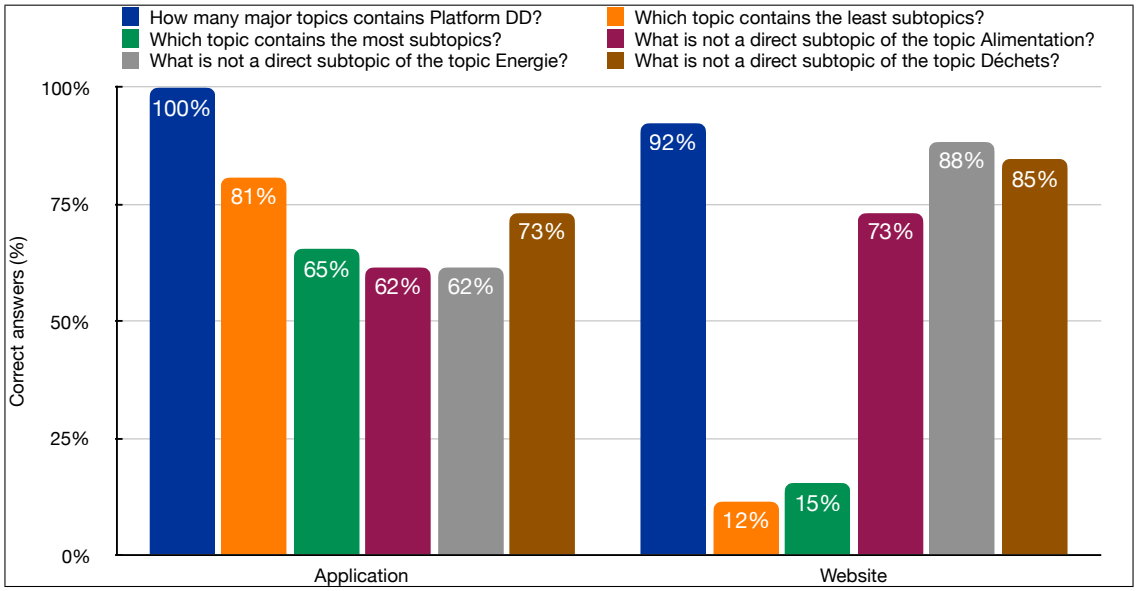


Figure 7.1: Results of the first 6 questions of part 1.

Figure 7.1 shows the results of the first six questions of the questionnaire. We can see that almost every participant correctly found that the content is divided over three major topics.

Participants using our application were significantly better in finding which topic contains the most and which the least number of subtopics. We expected this because in our application, the tree structure is more clear. As a consequence, these users have a better overview than the participants using the website.

To the questions about which topic is not a direct subtopic of a certain topic, the participants using the website gave a correct answer more frequently. This is probably because these users cannot see the indirect subtopics of the topics in the questions. Participants using our application possibly made some indirect subtopics visible by clicking on the plus-button.

7.2.3.2 Search & Find

The results to search and find related questions can be found in Figure 7.2. In general, we see that participants using our application are better in retrieving the information than the ones using the website. However, in one question participants using the website were slightly better in finding the correct result, but the difference is almost negligible.

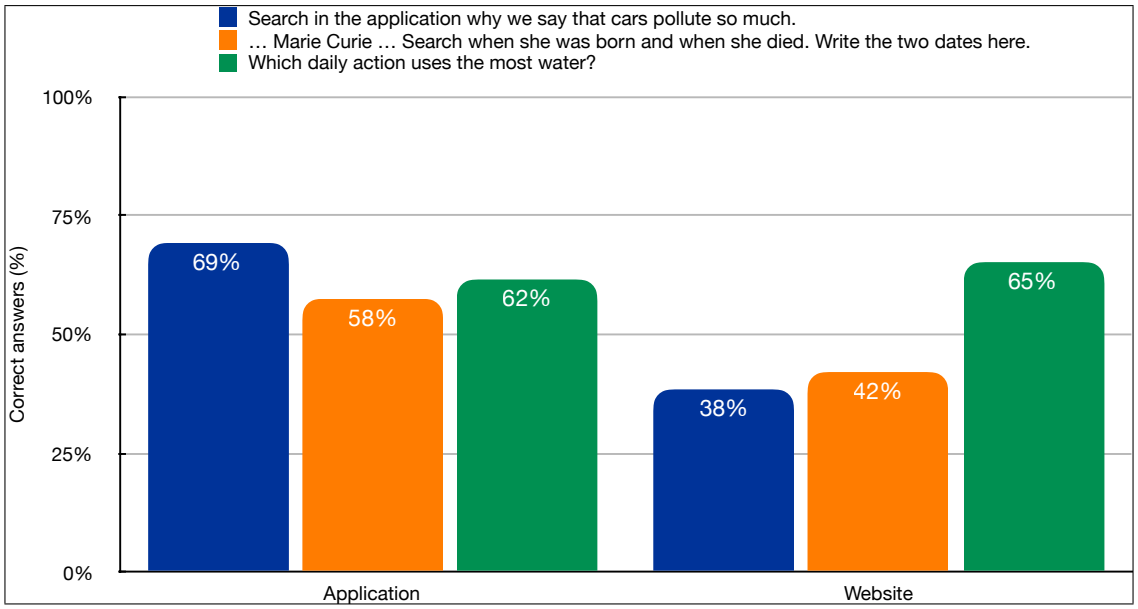


Figure 7.2: Results of search & find questions.

We also asked them to write as much titles as possible of topics that are still "under construction". This was not very successful because more than half of the participants could not find such a topic. The average number of topics found per participant is 0.42 in the case of our application and 0.81 in case of the website. This is probably because going through the topics on the website is quite fast, i.e. you just click on other topics until you find one that is under construction, while in our application you have to open and close the nodes one by one which takes more time.

In general, when looking at Figure 7.1 and Figure 7.2, we can conclude that our hypothesis was correct, i.e. participants using our application answer more questions correctly. Furthermore, in case our application has more correct answers, it outperforms the website almost every time. Hence, our application certainly is better than the website for structure comprehension (section 7.2.3.1) and searching and finding information.

7.2.3.3 Understanding of the Visualization

The participants using the application had to answer four additional (multiple choice) questions to test their understanding of the visualization. The results (Figure 7.3) make clear that most of the participants understand that subtopics give more detailed information about their parent node when a concrete example is used. However, when this question was formulated in a more general way (question B), a lot of the participants were wrong or indicated they did not know the answer. Even though the answer was almost revealed in question D (*A click on the plus-button of a bubble allows me to find projects on more specialized topics for that bubble*), only 35% found the correct answer.

Many participants did not know the answer on question C, which was about the

cross references (*There is no connection (arrow) between the bubble "Les déchets dans la nature" and the bubble "La pollution atmosphérique". However, are these topics in some other way related?*). Only 38% of the answers on this question was correct.

Hence, we were not very successful in testing the understanding of the visualization. There are two possible explanations. Either, the youngsters did not understand the visualization or the level of abstraction used in those questions was too high for them.

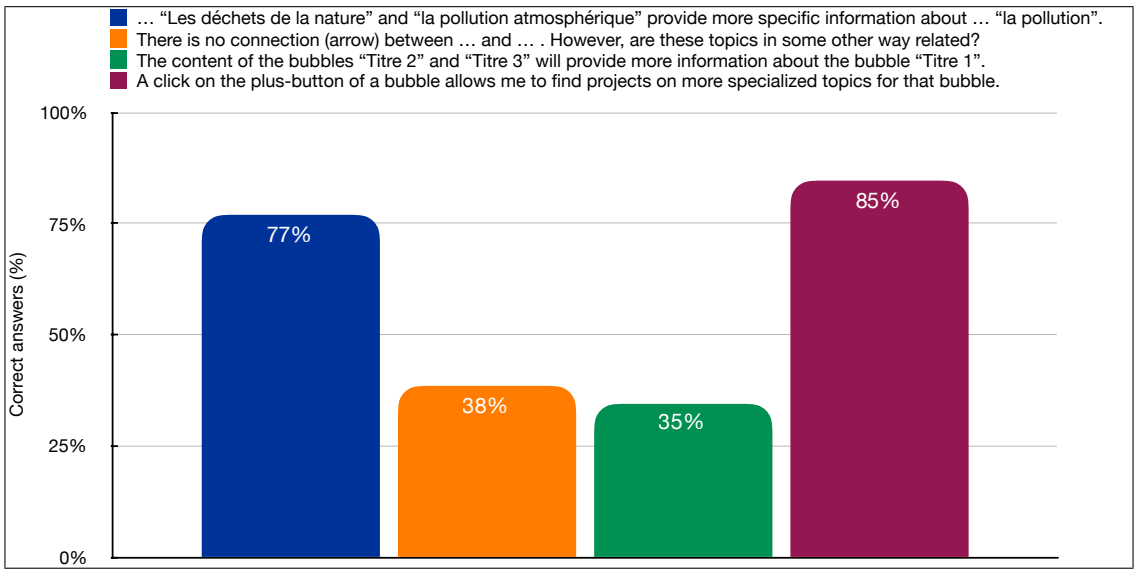


Figure 7.3: Results of questions about understanding of the visualization.

7.2.3.4 Usability

For the second part, we hypothesize that the website is less usable than our visualization and that our application provides a better user experience.

Figure 7.4 shows eight graphs, each representing the results of one of the eight questions about usability of Plateforme DD. When we look at the results of the first question, we can conclude that most of the participants (in both groups) are not very sure about whether they want to use the system more frequently or not.

In question two and three, we asked about the simplicity of the website or our application. We see that the majority of the participants using our application (strongly) disagrees with the statement that the tool is too difficult to use. On the other hand, we see a peak of participants using the website who agree with the same statement. In the next statement, the opposite was asked. Here we can make similar conclusions. In case of our application, the majority answered this question with "strongly agree", while in case of the website an equal percentage answered "disagree". Hence, based on these two questions, we can conclude that our application was easier to use than the website.

Furthermore, almost all participants using our application (all except 6%) think they will be able to use the tool in the future without help. In case of the website, 42% is neutral or disagrees with this statement. The fifth question enforces this result even more: 82% of the respondents using our application thinks their friends would be able to learn the system quickly, while in case of the website this is only 49%.

We see that participants using our application agree quite often with the sixth question, "I found it too much hassle to use the system", while the majority using the website disagrees with this statement. This is probably because our application was slower than expected on the used tablets. As a consequence, the tool was not always responding immediately, which was sometimes confusing for the users.

From the results of the seventh question, we can see that participants using the website were a bit more self-assured than participants using our application. In the last question, the distribution of the answers was similar for both systems, i.e. the majority did not need a lot of time before they could use the system smoothly.

Overall, based on the answers from this usability questionnaire, we can conclude that our application scores slightly better on usability than the website. Hence, by creating the tool we succeeded in our intention to make the structure of the content more clear and more attractable than the website. This is supported by a note on the form of a participant using the website: "I don't understand it".

7.2.3.5 User Experience

To evaluate the user experience, we used the UEQ questionnaire[4] (Appendix D) composed in such a way that inconsistencies can be easily detected. Together with the questionnaire, an excel-file is available that allows to analyze the answers and to create graphs.

The questionnaire consists of 26 terms at one side with their opposite at the other side, e.g. annoying versus enjoyable. A number between one and seven is used to give a score to this criterion. Each couple of terms, i.e. criterion, corresponds to one of six categories: attractiveness, perspicuity, efficiency, dependability, stimulation, and novelty.

The excel-file converts the numbers from one to seven to a range between -3 and +3, where -3 stands for the most negative answer and +3 for the most positive answer of the two. For each couple of opposites, the mean is calculated based on the converted answers, resulting in a number between -3 and +3. Because it is very unlikely to have a mean lower than -2 and higher than 2, the graphs are given with a range between -2 and +2. The resulting graphs can be found in Figure 7.5.

[4] `https://www.ueq-online.org/`

We can see that the score for the perspicuity of our application is higher than the one for the website, which means the application is more clear. A remarkable result is the one for efficiency: the application seems to be less efficient than the website. Maybe this can be explained as follows. The majority of the respondents indicated that our application was slow. As already mentioned, this is due to the devices used, but it was very inconvenient for them and this probably influenced the score for efficiency.

Further, we also see that our application scores significantly lower for novelty than the website. We think this is mainly due to the answers on the "dull/creative"-criterion. Participants using the website found the website more creative than the respondents using our application, which has its impact on the score for novelty.

The score for stimulation is average because the criteria concerning stimulation most of the time got scores between three and five. The score for dependability is a bit higher in case of the website. This is probably because our application was quite slow on the tablets, which had as a consequence that it not always met expectations.

Finally, the attractiveness of the website seems to be slightly higher. However, this difference is quite small and almost negligible.

Note that the excel-file also detects inconsistencies in the answers. The answers containing three or more inconsistencies (15 in total) were removed, as advised in the file. In this way, we should have the most correct results from the answers.

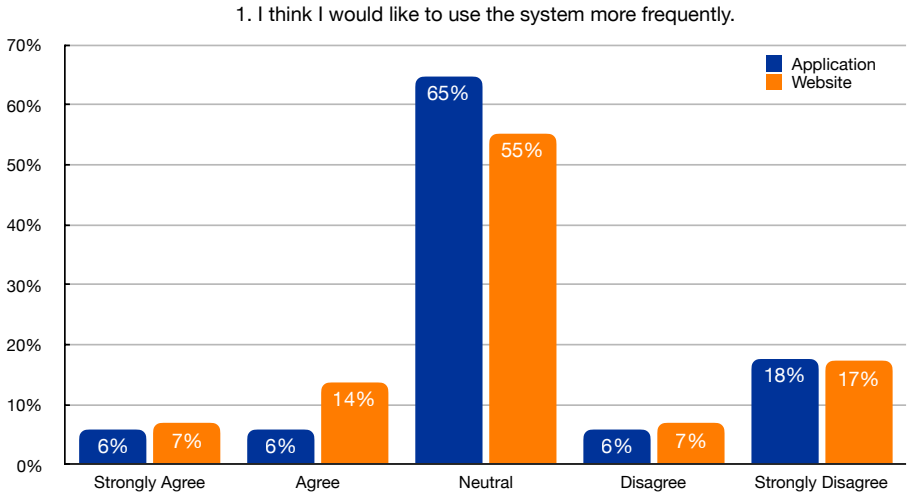
7.2.3.6 Overall

Our hypothesis that participants using our application answer more of part 1 correctly than the participants using the website is confirmed by the results.

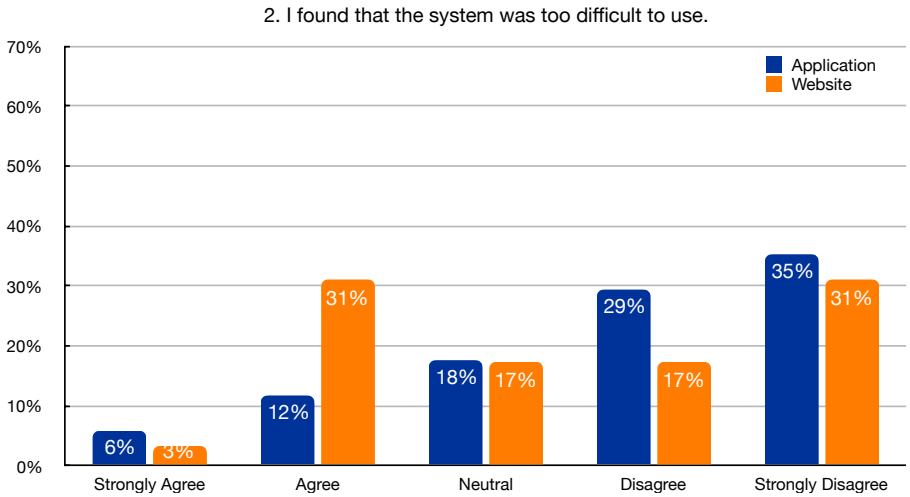
Further, our application scores slightly better on usability than the website. However, the questions testing the understanding of the visualization were not very well answered. Also, some questions about structure comprehension were answered better by the participants using the website. But for other questions, the results of participants using our application outperformed the other results.

The question to search for as much topics as possible that are "under construction" showed that the website allows users to browse faster through the topics. In our application, you always have to open and close nodes, which slows down the browsing process.

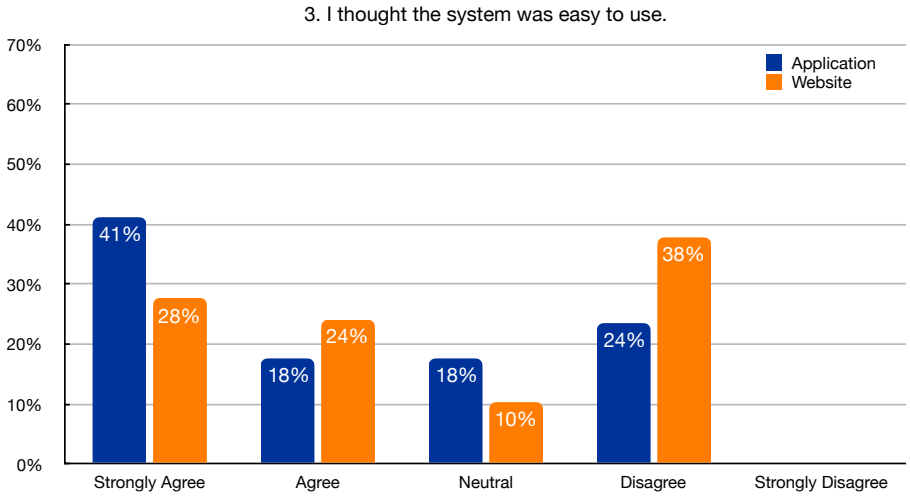
As a summary, we can say that our application makes the structure of the content more clear and is easier to use than the website, but some adaptations could still improve the visualization.



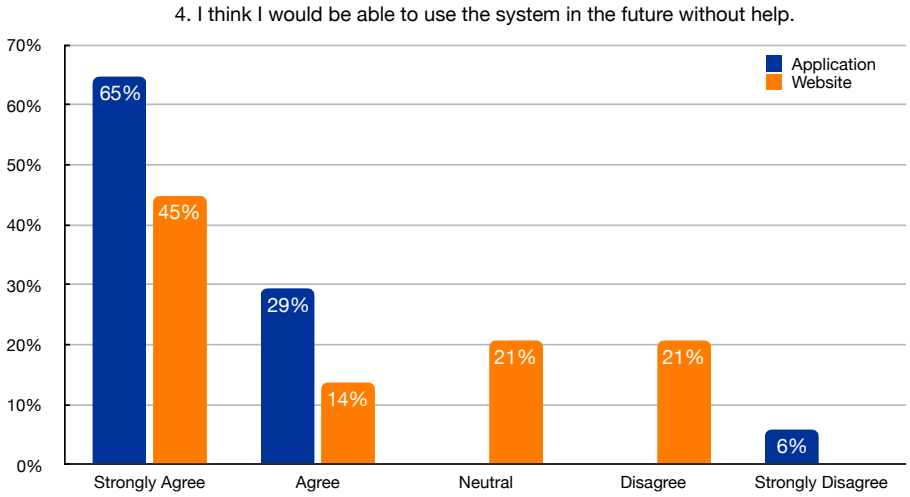
(a) Results for question 1.



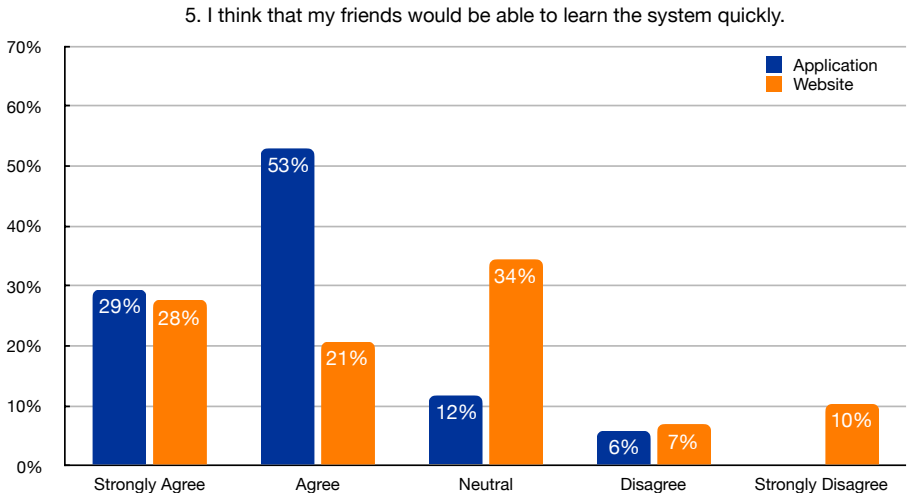
(b) Results for question 2.



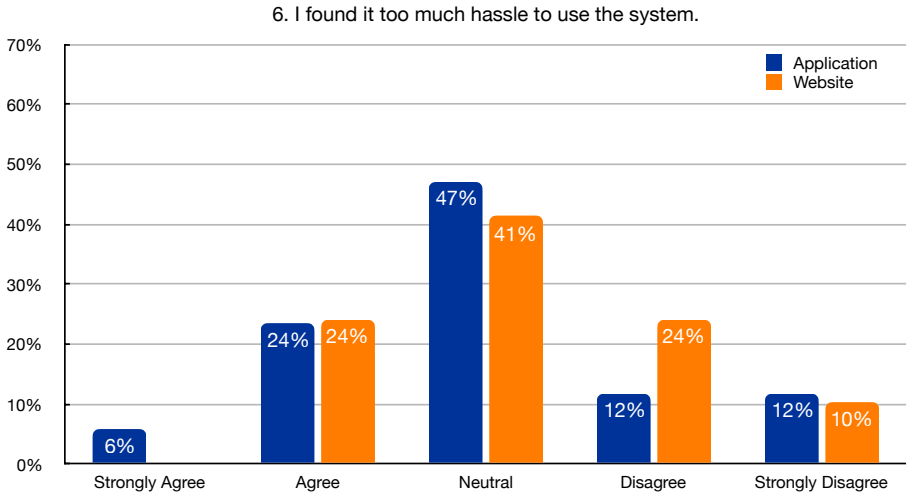
(c) Results for question 3.



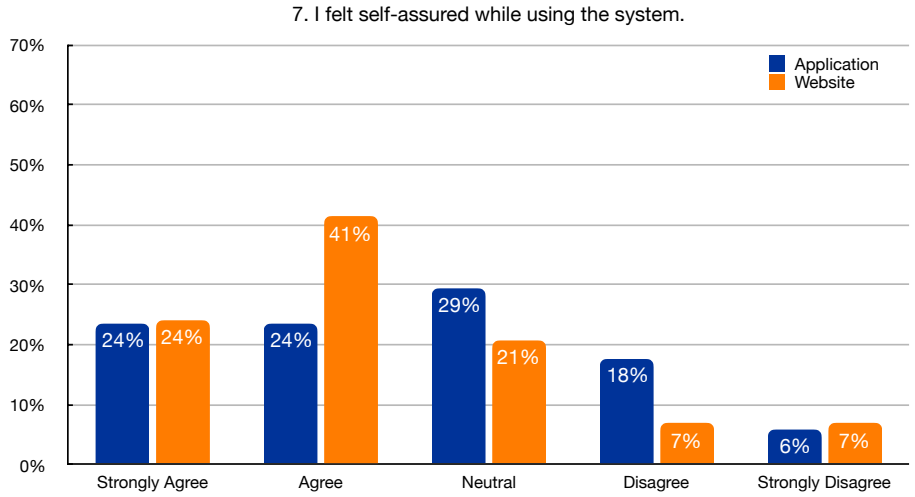
(d) Results for question 4.



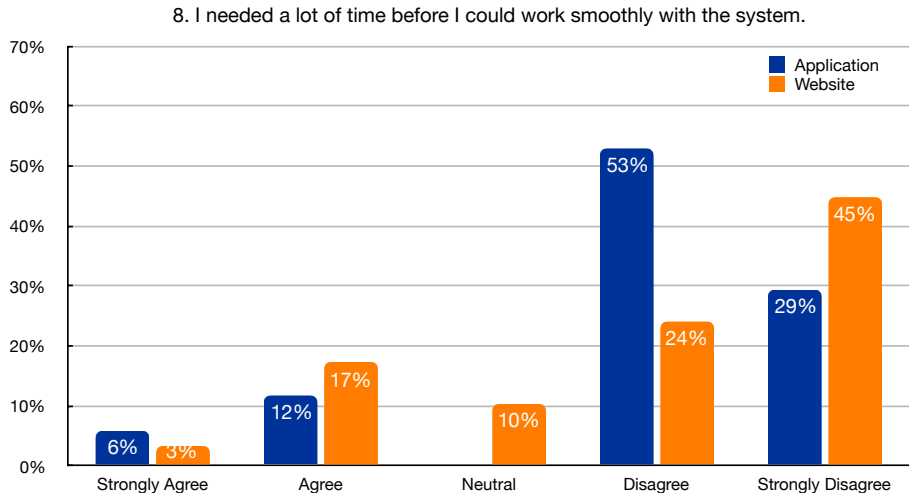
(e) Results for question 5.



(f) Results for question 6.

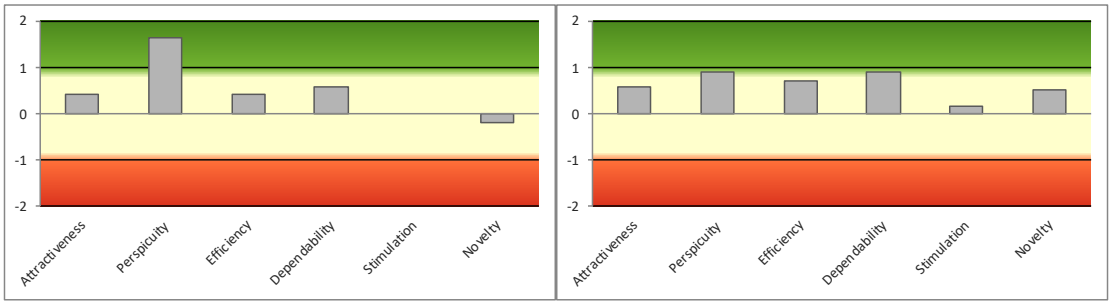


(g) Results for question 7.



(h) Results for question 8.

Figure 7.4: Usability Results Plateforme DD.



(a) Results User Experience Application (b) Results User Experience Website

Figure 7.5: User Experience Results Plateforme DD.

8 Conclusion & Future Work

8.1 Conclusion

In this thesis, we examined different visualization techniques in multiple domains. We started with a discussion of existing techniques and tools. It was clear that these were not very suitable for the goals we had in mind. We wanted a tool able to represent linked data and knowledge in a visual manner. Therefore, GuideaMaps was a good starting point. However, GuideaMaps has a number of limitations. First, the tool was created for iPads only. Hence, users are limited to a specific device and operating system. Further, there is no way to change the layout of the nodes and links. This thesis presents a solution to these and other limitations, called GuideaMaps 2.0.

To achieve the goals (section 1.2) and requirements (section 4.2), some important design decisions were made. The goal to have a device- and OS-independent application was achieved by the choice for a browser-based solution. In this way, users are not restricted to a particular device or OS anymore. Note that Google Chrome is currently the only browser in which the tool is tested, so other browsers can have a different (unexpected) behavior.

Another difference with the first version of GuideaMaps is that we made it easier to create template maps. In the first version, this had to be done with XML, while we now have a graphical way to do this. This feature should make it easier for users with less background in Computer Science to create templates.

Further, we proposed to make the tool collaborative such that multiple users could work on the same visualization at the same time (requirement FR 9). However, this functionality is not yet provided in GuideaMaps 2.0.

The biggest contribution of this thesis can be found in the genericity of our application. The tool can be used for much more than only one domain (e.g. requirement elicitation). To achieve this, our application is created as a library. Developers can reuse the core of our application and replace the code responsible for the visualization of the nodes and links by their own implementation. The code can simply be plugged in without affecting the general implementation.

We strongly believe that this is an important feature and tested the ease to plug in other code by means of a use case, i.e. Plateforme DD. It takes only a minute to plug in a custom implementation and run the library. We also took the opportunity to perform a user experiment with this use case. Half of the participants worked with a website with a tree structure under the hood, while the other half worked with the same content of the website but visualized by our application. The intent of visualizing the content was to make the structure more understandable. The results of the questionnaire confirmed our expectations: searching and finding information is easier when using our application than when using the website. However, browsing through the topics is faster on the website than in our application and the understanding of the visualization was not very successful.

8.2 Future Work

Next to fixing some usability issues as detected by the user studies, the presented application could be improved in different ways in the future. One of the most important improvements that could be done is supporting a broader range of browsers. Currently, the tool works well in the latest version of Google Chrome. It is good to have a working tool in one of the most widely used browsers, but lots of people use other browsers apart from Google Chrome. We detected some issues in Firefox, where for instance the tool is very slow on a tablet. In the future, we should improve the system such that it works in different browsers and if possible in all commonly used ones.

Furthermore, it is essential to be able to store the created templates and maps. Therefore, the system should be further elaborated so that the changes are written to a file that can be loaded the next time the map is opened. Right now, the maps are re-initialized every time the page is refreshed.

Next to supporting persistence, it would be useful to make the tool collaborative, i.e. allow multiple people to work simultaneously on the same map and see each others (saved) changes in real-time. This was formulated as requirement FR 9, but is not supported in the current version.

Another important improvement would be to make it possible to drag and drop the nodes of the visualization to support the similarity and proximity principle of the Gestalt Psychology Theory (Koffka, 2013).

References

Balaid, A., Abd Rozan, M. Z., Norris Hikmi, S., & Memon, J. (2016). Knowledge maps: A systematic literature review and directions for future research. *International Journal of Information Management*, *36*, 451–475. doi: 10.1016/j.ijinfomgt.2016.02.005

Cañas, A. J., Hill, G., Carff, R., Suri, N., Lott, J., Eskridge, T., ... Carvajal, R. (2004). CmapTools: A Knowledge Modeling and Sharing Environment. In *Concept Maps: Theory, Methodology, Technology. Proc. of the First Int. Conference on Concept Mapping* (pp. 125–135). Universidad Pública de Navarra, Pamplona, Spain.

Davies, M. (2011). Concept Mapping, Mind Mapping and Argument Mapping: What are the Differences and Do They Matter? *High Educ*, *62*, 279–301. doi: 10.1007/s10734-010-9387-6

De Troyer, O., & Janssens, E. (2014, August). A feature modeling approach for domain-specific requirement elicitation. In *Requirements Patterns (RePa), 2014 IEEE 4th International Workshop on* (pp. 17–24). IEEE. doi: 10.1109/RePa.2014.6894839

Eppler, M. J. (2006). A comparison between concept maps, mind maps, conceptual diagrams, and visual metaphors as complementary tools for knowledge construction and sharing. *Information Visualization*, *5*(3), 202–210. Retrieved from `http://dblp.uni-trier.de/db/journals/ivs/ivs5.html#Eppler06`

Goldenberg, J., Mazursky, D., & Solomon, S. (1999). Toward Identifying the Inventive Templates of New Products: A Channeled Ideation Approach. *Journal of Marketing Research*, *36*(2), 200–210. doi: 10.2307/3152093

ISO, E. (1998). *ISO 9241/11: Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs) - Part 11: Guidance on Usability*.

Janssens, E. (2013). *Supporting Requirement Elicitation for Serious Games using a Guided Ideation tool on iPad.* Master Thesis Vrije Universiteit Brussel. Retrieved from `https://wise.vub.ac.be/sites/default/files/theses/thesis-ErikJanssens.pdf`

Koffka, K. (2013). *Principles Of Gestalt Psychology*. Taylor & Francis. doi: 10.4324/9781315009292

Moody, D. (2009, November). The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. *IEEE Transactions on Software Engineering*, *35*(6), 756–779. Retrieved from `https://doi.org/10.1109/TSE.2009.67` doi: 10.1109/TSE.2009.67

Nesbit, J. C., & Adesope, O. O. (2006). Learning With Concept and Knowledge Maps: A Meta-Analysis. *Review of Educational Research*, *76*(3), 413–448. doi: 10.3102/00346543076003413

Novak, J. D., & Cañas, A. J. (2006). The Origins of the Concept Mapping Tool and the Continuing Evolution of the Tool. *Information Visualization*, *5*(3), 175–184. doi: 10.1057/palgrave.ivs.9500126

O'Donnell, A. M., Dansereau, D. F., & Hall, R. (2002). Knowledge Maps as Scaffolds for Cognitive Processing. *Educational Psychology Review*, *14*(5). doi: 10.1023/A:1013132527007

Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., & Carey, T. (1994). *Human-computer interaction*. Essex, UK, UK: Addison-Wesley Longman Ltd.

Scherl, A., Dethleffsen, K., & Meyer, M. (2012). Interactive knowledge networks for interdisciplinary course navigation within moodle. *Advances in Physiology Education*, *36*(4), 284–297. doi: 10.1152/advan.00086.2012

Shih, P. (2011). *Brainstorming Beyond the Laboratory: Idea Generation Practices in Software Development Firms*. University of California.

Susi, T., Johannesson, M., & Backlund, P. (2007). *Serious Games: An Overview*. Institutionen för kommunikation och information (Sweden).

Tullis, T. S., & Stetson, J. N. (2004). A Comparison of Questionnaires for Assessing Website Usability. In *Usability Professional Association Conference (UPA)* (Vol. 1).

Wiegmann, D., F. Dansereau, D., C. McCagg, E., L. Rewey, K., & Pitre, U. (1992). Effects of Knowledge Map Characteristics on Information Processing. *Contemporary Educational Psychology*, *17*, 136–155. doi: 10.1016/0361-476X(92)90055-4

A

GM Evaluation Map Creator

Thesis evaluation

The goal of my thesis was to create a tool, similar to mind-maps, to create and show a visualization of linked data. The difference with existing mind-map tools is that the structure of the map (called template) can be pre-defined.
Therefore, the tool has two modes.
(1) Map creator mode: A map creator can create new templates. His task is to create the necessary nodes and links, containing a description about the expected content for the particular nodes.
(2) End-user mode: End-users can create a map by starting from a template and providing content. They also can extend the map with new nodes in a limited way.

An example of a template can be found on http://thijs-evaluation.surge.sh/. This template is intended to be used for collecting the requirements for an e-commerce website.

Note that there are two different types of nodes: regular nodes (as in a mind-map) and choice nodes. The choice nodes contain pre-defined options, from which the end-user can select one or more options. For example, there could be a "gender" node with two options: "male" and "female". In this case, it can be specified that the end-user should only select one option.
Next, nodes can be optional, which means that an end-user may decide not to use them. For instance, an e-commerce website may contain a user profile, but for simple website this may not be needed. Therefore, this node is optional (indicated by a dotted link), meaning that the end-user can deselect the node.

Via this experiment, I want to evaluate the map creator part of my thesis. I will give you some tasks to perform and after each task you have to indicate how intuitive it was to perform the task , i.e. how easy it was to find the relevant operations in the application and execute them.

At the end, a general Usability Questionnaire is added to test the overall usability of my tool and you can give your opinion about the tool and its functionality.

I assume that you will need 10-15 minutes to finish this questionnaire.

*Required

Before we start

Before we start with the actual evaluation, we want to know a bit more about who you are. We only need this information to compare the results of people with the same characteristics. All information is used anonymously.

1. **What is your age?** *

2. **Do you have a background in Computer Science and/or visualization techniques?** *
   Mark only one oval.

   - ◯ Yes
   - ◯ No

Tasks

To start, go to http://thijs-evaluation.surge.sh/ in Google Chrome. By opening the link, you will see the "e-commerce" template. This template is intended to be used for collecting the requirements for an e-commerce website.

The first thing you have to do is to make sure that you are in the map creator mode. Because there is not yet a login functionality in our system, you can select the mode at the top left. Click on the "map creator" button. You may take some time to inspect the template.

Now we want you to try to create another (similar but smaller) template for e-commerce. To do that, select "New map" in the middle menu at the top.

3. **Provide a title to the root node and name it "e-commerce". How intuitive was it to perform this action?** *
   Mark only one oval.

   - ◯ Very intuitive
   - ◯ Intuitive
   - ◯ Neutral
   - ◯ Not intuitive
   - ◯ Absolutely not intuitive

4. **Add a regular (mandatory) node with title "Product" and give it the description "Add here the products for sale". How intuitive was it to perform this action?** *
   Mark only one oval.

   - ◯ Very intuitive
   - ◯ Intuitive
   - ◯ Neutral
   - ◯ Not intuitive
   - ◯ Absolutely not intuitive

5. **Add a mandatory choice node to the "e-commerce" node with title "Searching" and the description "Select here how you will allow users to search in the website". This choice node should have two options: "search" and "browse". The user should be allow to select both options. How intuitive was it to perform this action?** *

Mark only one oval.

- ◯ Very intuitive
- ◯ Intuitive
- ◯ Neutral
- ◯ Not intuitive
- ◯ Absolutely not intuitive

Your template should now look like this. If not, redo the previous steps to make sure you end up with this visualization.

6. **Add an additional option (the title of it does not matter) to the choice node "searching". How intuitive was it to perform this action?** *

Mark only one oval.

- ◯ Very intuitive
- ◯ Intuitive
- ◯ Neutral
- ◯ Not intuitive
- ◯ Absolutely not intuitive

7. **Now assume the "browse"-option is wrong and you want to delete it so that end-users cannot select it later on. Try to delete this option. How intuitive was it to perform this action?** *

Mark only one oval.

- ◯ Very intuitive
- ◯ Intuitive
- ◯ Neutral
- ◯ Not intuitive
- ◯ Absolutely not intuitive

8. **Now assume the "Product" node is wrong and you want to delete it. Try to delete this node. How intuitive was it to perform this action?** *

Mark only one oval.

- Very intuitive
- Intuitive
- Neutral
- Not intuitive
- Absolutely notintuitive

9. **Now add a regular optional node with title "Profile". How intuitive was it to perform this action?** *

Mark only one oval.

- Very intuitive
- Intuitive
- Neutral
- Not intuitive
- Absolutely notintuitive

10. **Make a screenshot of the template you created and upload it here.**

Usability Questionnaire

The first part of the questionnaire is now finished. Please answer the next 10 (small) questions about the general usability of the application. These are the default questions of the SUS questionnaire.

11. **1. I think I would like to use the system frequently. (Suppose that such a tool could be useful for your work, study or hobbies).** *

Mark only one oval.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

12. **2. I found the system unnecessarily complex.** *

Mark only one oval.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

13. **3. I thought the system was easy to use.** *

Mark only one oval.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

14. **4. I think that I would need the support of a technical person to be able to use the system.** *

Mark only one oval.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

15. **5. I found the various functions in the system were well integrated.** *

Mark only one oval.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

16. **6. I thought there was too much inconsistency in this system.** *

Mark only one oval.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

17. **7. I would imagine that most people would learn to use the system very quickly.** *

Mark only one oval.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

18. 8. I found the system very cumbersome to use. *

Mark only one oval.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

19. 9. I felt very confident using the system. *

Mark only one oval.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

20. 10. I needed to learn a lot of things before I could get going with this system. *

Mark only one oval.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

Almost finished...

Before we finish, I really like to know your opinion about the tool.

21. Please tell me what you liked about the app, what was good?

22. What could have been done better?

23. Thanks a lot for finishing this questionnaire. You helped me a lot for the evaluation of my thesis. If you have any remarks, please write them down here.

B

GM Evaluation End User

Thesis evaluation

The goal of my thesis was to create a tool, similar to mind-maps, to create and show a visualization of linked data. The difference with existing mind-map tools is that the structure of the map (called template) can be pre-defined.
Therefore, the tool has two modes.
(1) Map creator mode: A map creator can create new templates. His task is to create the necessary nodes and links, containing a description about the expected content for the particular nodes.
(2) End-user mode: End-users can create a map by starting from a template and providing content. They also can extend the map with new nodes in a limited way.

An example of a template can be found on http://thijs-evaluation.surge.sh/. This template is intended to be used for collecting the requirements for an e-commerce website.

Note that there are two different types of nodes: regular nodes (as in a mind-map) and choice nodes. The choice nodes contain pre-defined options, from which the end-user can select one or more options. For example, there could be a "gender" node with two options: "male" and "female". In this case, it can be specified that the end-user should only select one option.
Next, nodes can be optional, which means that an end-user may decide not to use them. For instance, an e-commerce website may contain a user profile, but for simple website this may not be needed. Therefore, this node is optional (indicated by a dotted link), meaning that the end-user can deselect the node.

Via this experiment, I want to evaluate the end-user part of my thesis. I will give you some tasks to perform and after each task you have to indicate how intuitive it was to perform the task , i.e. how easy it was to find the relevant operations in the application and execute them.

At the end, a general Usability Questionnaire is added to test the overal usability of my tool and you can give your opinion about the tool and its functionality.

I assume that you will need 10-15 minutes to finish this questionnaire.

*Required

Before we start

Before we start with the actual evaluation, we want to know a bit more about who you are. We only need this information to compare the results of people with the same characteristics. All information is used anonymously.

1. **What is your age?** *

2. **Do you have a background in Computer Science and/or visualization techniques?** *
   Mark only one oval.

   - ◯ Yes
   - ◯ No

Tasks

To start, go to http://thijs-evaluation.surge.sh/ in Google Chrome. By opening the link, you will see the "e-commerce" template. This template is intended to be used for collecting the requirements for an e-commerce website.

The first thing you have to do is to make sure that you are in the end-user mode. Because there is not yet a login functionality in our system, you can select the mode at the top left. Click on the "end-user" button if not yet selected.

3. **Add content to the "Product" node by writing what information you want to provide about a product in its content-field (for example, the price, a description, ...). How intuitive was it to perform this action?** *
   Mark only one oval.

   - ◯ Very intuitive
   - ◯ Intuitive
   - ◯ Neutral
   - ◯ Not intuitive
   - ◯ Absolutely not intuitive

4. **If you look at the bullet next to the title of the "Product" node, you will see it is not the same as this bullet in other nodes. One of them is half full, another is full and others are empty. Do you understand the meaning of the bullet next to the title of a node? If so, explain it here.** *

5. **The template doesn't allow to enter information about social media. Therefore, add a regular, non-optional node with the title "Social Media" and with description "Enter the social media you want to support" to the "e-commerce" node. How intuitive was it to perform this action?** *

Mark only one oval.

- Very intuitive
- Intuitive
- Neutral
- Not intuitive
- Absolutely not intuitive

6. **Now indicate that you want to allow the users of your website to search for information by means of a regular search as well as by means of browsing. You can do this by means of the choice node with title "Searching". If you did it correctly, your selected options should be added as child nodes of the Searching-node. How intuitive was it to perform this action?** *

Mark only one oval.

- Very intuitive
- Intuitive
- Neutral
- Not intuitive
- Absolutely not intuitive

7. **Find a way to delete one of these options, for instance the Search-option. How intuitive was it to perform this action?** *

Mark only one oval.

- Very intuitive
- Intuitive
- Neutral
- Not intuitive
- Absolutely not intuitive

8. **Find a way to add a new option, called "Via Facebook" to the "Registration" choice node. How intuitive was it to perform this action?** *

Mark only one oval.

- Very intuitive
- Intuitive
- Neutral
- Not intuitive
- Absolutely not intuitive

9. **The node "Contact Us" has a lock symbol. Explain for which purpose you think you can use this.** *

10. **Click on the zoom-to-fit button in the top-right corner and make a screenshot of the visualization. Please upload that screenshot here.**

Usability Questionnaire

The first part of the questionnaire is now finished. Please answer the next 10 (small) questions about usability. These are the default questions of a SUS questionnaire.

11. **1. I think I would like to use the system frequently. (Suppose that such a tool could be useful for your work, study or hobbies).** *

Mark only one oval.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

12. **2. I found the system unnecessarily complex.** *

Mark only one oval.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

13. **3. I thought the system was easy to use.** *

Mark only one oval.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

14. **4. I think that I would need the support of a technical person to be able to use the system.** *

Mark only one oval.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

15. **5. I found the various functions in the system were well integrated.** *

Mark only one oval.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

16. **6. I thought there was too much inconsistency in this system.** *

Mark only one oval.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

17. **7. I would imagine that most people would learn to use the system very quickly.** *

Mark only one oval.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

18. **8. I found the system very cumbersome to use.** *

Mark only one oval.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

19. 9. I felt very confident using the system. *

Mark only one oval.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

20. 10. I needed to learn a lot of things before I could get going with this system. *

Mark only one oval.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

Almost finished

Before we finish, I really like to know your opinion about the tool.

21. Please tell me what you liked about the app, what was good?

22. What could have been done better?

23. Thanks a lot for finishing this questionnaire. You helped me a lot for the evaluation of my thesis. If you have any remarks, please write them down here.

C

Plateforme DD Questionnaire

O Male O Female Age: Group: Application / Website

PART 1

1. How many major topics contains the DD platforme?
   1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10

2. Which topic contains the least subtopics?
   Alimentation - Energie - Déchets

3. Which topic contains the most subtopics?
   Alimentation - Energie - Déchets

4. What is not a direct subtopic of the topic **Alimentation?**
   La filière alimentaire - Le gaspillage alimentaire -
   La transformation alimentaire - L'agriculture - L'assiette

5. What is not a direct subtopic of the topic **Energie?**
   La production énergétique - La distribution énergétique -
   L'énergie et la société - La consommation énergétique - Les formes d'énergie

6. What is not a direct subtopic of the topic **Déchets?**
   Le tri des déchets - Les déchets dans la nature -
   Le gaspillage alimentaire - Zéro déchets - Le gaspillage - La pollution

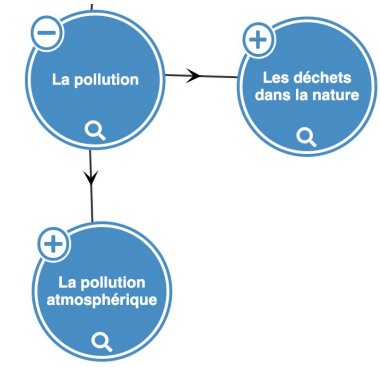
1. Search in the application why we say that cars pollute so much.

2. Write down as much titles of the pages “*en cours d'écriture*”.

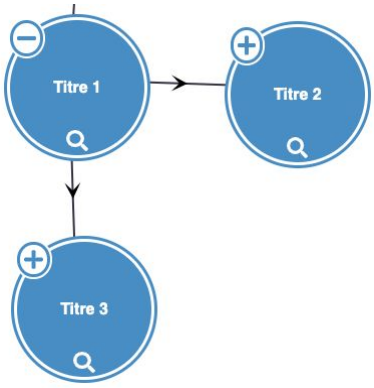
3. When talking about “la production énergétique”, Marie Curie did important research about radioactivity. Search when she was born and when she died. Write the two dates here.

4. Which daily action uses the most water?

5. For each of the following sentences, indicate the correct answer. If you don’t know the answer, select “I don’t know”



A. The content of the bubbles “Les déchets de la nature” and “La pollution atmosphérique” provide more specific information about pollution than the content of the bubble “La pollution”.
The sentence is correct - The sentence is not correct - I don’t know



B. The content of the bubbles “Titre 2” and “Titre 3” will provide more information about Titre 1.
The sentence is correct - The sentence is not correct - I don’t know

C. There is no connection (arrow) between the bubble “Les déchets dans la nature” and the bubble “La pollution atmosphérique”. However, are these topics in some other way related?
Yes - No - I don’t know

D. A click on the plus-button of a bubble allows me to find projects on more specialized topics for that bubble.
Correct - Not correct - I don’t know

PART 2

1. I think I would like to use the system more often.
   O Strongly agree
   O Agree
   O Neutral
   O Disagree
   O Strongly disagree

2. I found that the system was too difficult to use.
   O Strongly agree
   O Agree
   O Neutral
   O Disagree
   O Strongly disagree

3. I thought the system was easy to use.
   O Strongly agree
   O Agree
   O Neutral
   O Disagree
   O Strongly disagree

4. I think I would be able to use the system in the future without help.
   O Strongly agree
   O Agree
   O Neutral
   O Disagree
   O Strongly disagree

5. I think that my friends would be able to learn to use the system quickly.
   O Strongly agree
   O Agree
   O Neutral
   O Disagree
   O Strongly disagree

6. I found it too much hassle to use the system.
   O Strongly agree
   O Agree
   O Neutral
   O Disagree
   O Strongly disagree

7. I felt self-assured while using the system.
   O Strongly agree
   O Agree
   O Neutral
   O Disagree
   O Strongly disagree

8. I needed a lot of time before I could work smoothly with the system.
   O Strongly agree
   O Agree
   O Neutral
   O Disagree
   O Strongly disagree

D

User Experience Questionnaire

Please make your evaluation now.

For the assessment of the product, please fill out the following questionnaire. The questionnaire consists of pairs of contrasting attributes that may apply to the product. The circles between the attributes represent gradations between the opposites. You can express your agreement with the attributes by ticking the circle that most closely reflects your impression.

Example:

attractive	○	⊗	○	○	○	○	○	unattractive

This response would mean that you rate the application as more attractive than unattractive.

Please decide spontaneously. Don't think too long about your decision to make sure that you convey your original impression.

Sometimes you may not be completely sure about your agreement with a particular attribute or you may find that the attribute does not apply completely to the particular product. Nevertheless, please tick a circle in every line.

It is your personal opinion that counts. Please remember: there is no wrong or right answer!

Please assess the product now by ticking one circle per line.

	1	2	3	4	5	6	7		
annoying	○	○	○	○	○	○	○	enjoyable	1
not understandable	○	○	○	○	○	○	○	understandable	2
creative	○	○	○	○	○	○	○	dull	3
easy to learn	○	○	○	○	○	○	○	difficult to learn	4
valuable	○	○	○	○	○	○	○	inferior	5
boring	○	○	○	○	○	○	○	exciting	6
not interesting	○	○	○	○	○	○	○	interesting	7
unpredictable	○	○	○	○	○	○	○	predictable	8
fast	○	○	○	○	○	○	○	slow	9
inventive	○	○	○	○	○	○	○	conventional	10
obstructive	○	○	○	○	○	○	○	supportive	11
good	○	○	○	○	○	○	○	bad	12
complicated	○	○	○	○	○	○	○	easy	13
unlikable	○	○	○	○	○	○	○	pleasing	14
usual	○	○	○	○	○	○	○	leading edge	15
unpleasant	○	○	○	○	○	○	○	pleasant	16
secure	○	○	○	○	○	○	○	not secure	17
motivating	○	○	○	○	○	○	○	demotivating	18
meets expectations	○	○	○	○	○	○	○	does not meet expectations	19
inefficient	○	○	○	○	○	○	○	efficient	20
clear	○	○	○	○	○	○	○	confusing	21
impractical	○	○	○	○	○	○	○	practical	22
organized	○	○	○	○	○	○	○	cluttered	23
attractive	○	○	○	○	○	○	○	unattractive	24
friendly	○	○	○	○	○	○	○	unfriendly	25
conservative	○	○	○	○	○	○	○	innovative	26