Exclusive Design

Vasilis van Gemert

Exclusive Design

Let me start this thesis with exaggerating a bit: Roughly said, in the past 25 years we have been designing websites mostly for people who design websites. This means there is an incredible body of knowledge when it comes to designing for people who use their computers in a similar way as we do.

But if we want to create truly inclusive websites, expertise in ourselves is not enough. We also need expertise in designing interfaces for people who are excluded. This expertise is lacking. In this research at the Master Design at the Willem de Kooning Academy in Rotterdam I have worked with the question

> *What if we design websites exclusively for people with disabilities?*

In other words, what if we flip the so called *ability bias*, and start creating tailor made experiences for, and with real people with real disabilities?

I tried to find an answer to this question and its subquestions by working with a set of Exclusive Design Principles. I created these principles when I was studying a set of *inclusive design principles*. It turned out that the exact opposite of these inclusive principles was the perfect starting point for my research.[1]

The exclusive design principles are:

Study situation

I used this principle because I wonder if *we understand the different contexts of people with disabilities well enough?*[2] My assumption is that we don't. So in order to become a specialist inclusive designer I studied a few individual situations of people with different disabilities.

Ignore conventions

The original inclusive principle says that you should use conventions that people know. But I couldn't help but wonder: *"Do the current web design conventions work for people with disabilities?"*[3] Simply said, the current conventions are designed by, and thus for, designers. Not all these conventions work for non-designers. If we want to include non-designers, and especially people with disabilities, we should reconsider these conventions if needed, after we studied their situations.

Prioritise identity

Observing the situation of people with disabilities, and designing things especially for them is of course a good first step. But *what if we let people with disabilities play an active role in the design process?*[4] Next to designing *for* people I have also designed *with* people, combining the insights and ideas of excluded people with the skills and knowledge of me as a webdesigner.

Add nonsense

One of my main concerns is *how can we lift accessible web design beyond the functional?*[5] I have tried to answer this question by allowing the people I worked with, and myself, to add nonsense, in order to try and come up with ideas that live on a higher conceptual level than the obvious. This has resulted in some interesting — and fun — new ideas and projects.

Why?

For me the most important reason we should be designing inclusive websites is *because we can*. And the effects are big. Creating inclusive websites enables people with disabilities to lead a more independent life. This should be more than enough reason in itself. But there are other pressing reasons why we need expertise in inclusive design.

In Europe, including the Netherlands where I live and teach, we have signed and ratified the UN Convention on the Rights of Persons with Disabilities, and new laws are in place to enforce this. This means that all new government websites must be accessible. All of them, from small municipalities to large departments. This means that the design teams and design contractors that work on these websites *must* have accessibility expertise.

There is also a simple argument to be made that inclusive websites are a good idea from a business point of view. If more people can use your site, more people will use your product. And there are more financial reasons to design inclusively.

In the United States there is a culture of suing. Recently Beyoncé was sued because her website *violates the Americans With Disabilities Act*: It fails to accommodate visually impaired users. In the past these kinds of lawsuits have been settled for millions. In the Netherlands we don't have a suing culture, but the government *could* fine websites that break the new accessibility laws.

While indeed these are all valid arguments, I prefer the first, more positive reason to create inclusive websites. We should create inclusive websites because we can.

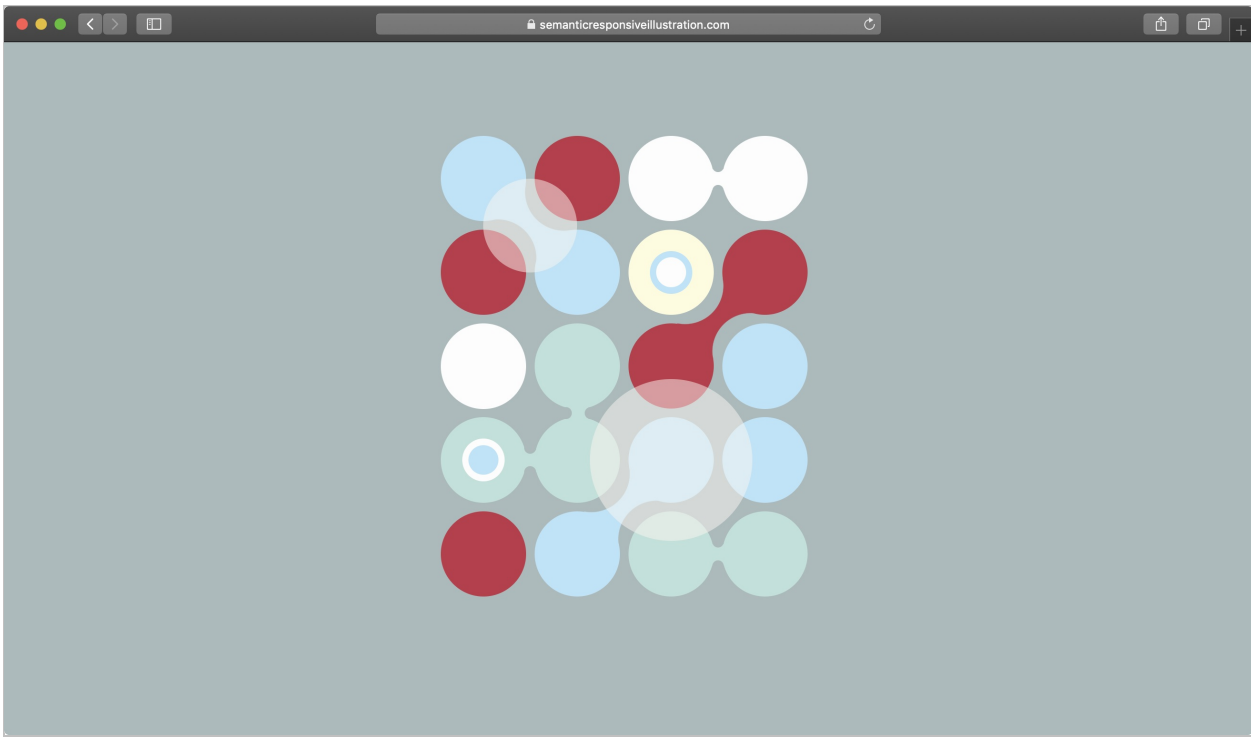
> And that's what I did. Or to be more precise, I have done the exact opposite. I have designed and created tailor made websites, exclusively for individual people with disabilities. But before I write about these websites I think I can better start with writing about the weird, weird web, and by

explaining *Everybody's paradox*.

Everybody’s paradox

I really like the web. I like it because it is the only medium that’s designed to really work for everyone. In my conference talks I often show my enthusiasm for the web with an example of an illustration of circles that I created using web technology.

An illustration for everybody

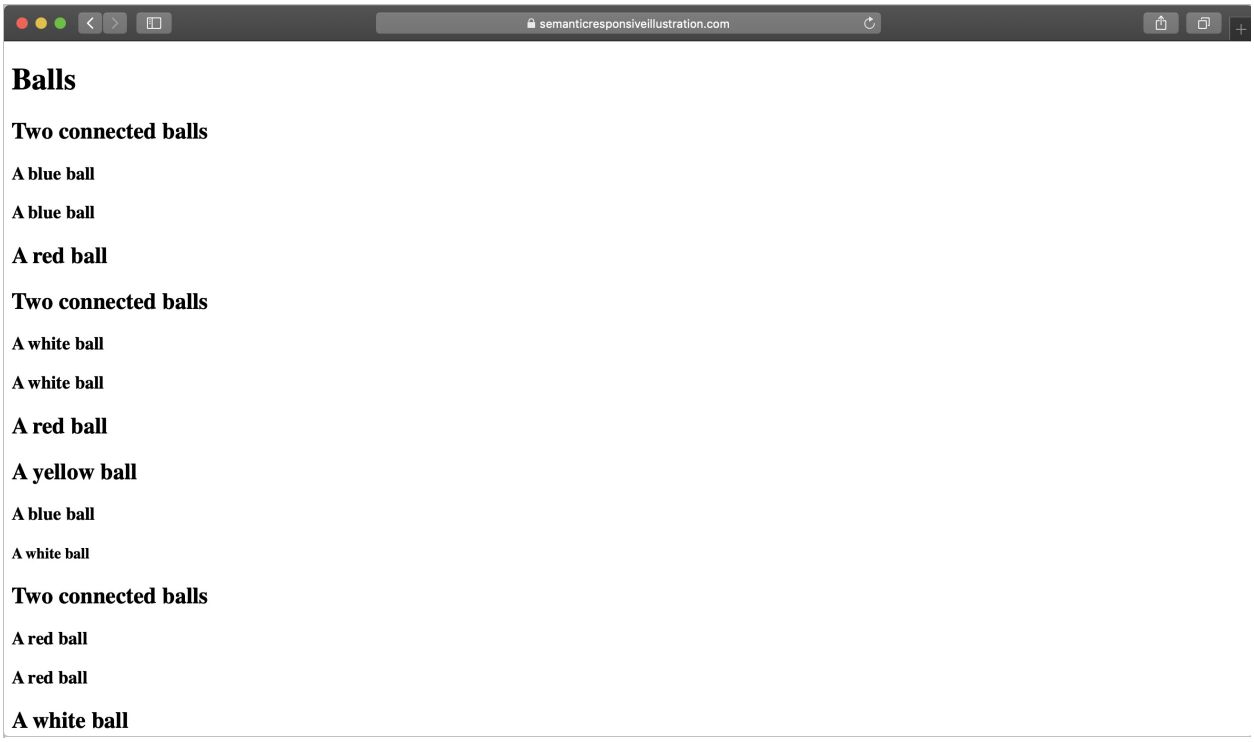


The illustration of coloured circles shown in a browser

The illustration can be seen by everybody who has eye sight and a browser. It can even adapt to different contexts, like screen sizes.

View this video online: vvg.gr/qz

But there are other contexts the web can deal with as well. For instance, not all browsers support the same set of features. Some older browsers don't support generating rounded corners. In this case you could consider to present the user with a textual representation of the image instead.



A textual version of the illustration, which reads: Balls, Two connected balls, a blue ball, a blue ball, a red ball, two connected balls, a white ball, a white ball …

And this brings us to yet another context: the web can be used by people who can see, but it can also be used by people who cannot see. These people may depend on assistive technology that reads what's on the screen.

I really like the web because of this incredible feature: the web can be used by everybody, regardless of the software and hardware you use.[6]

In theory.

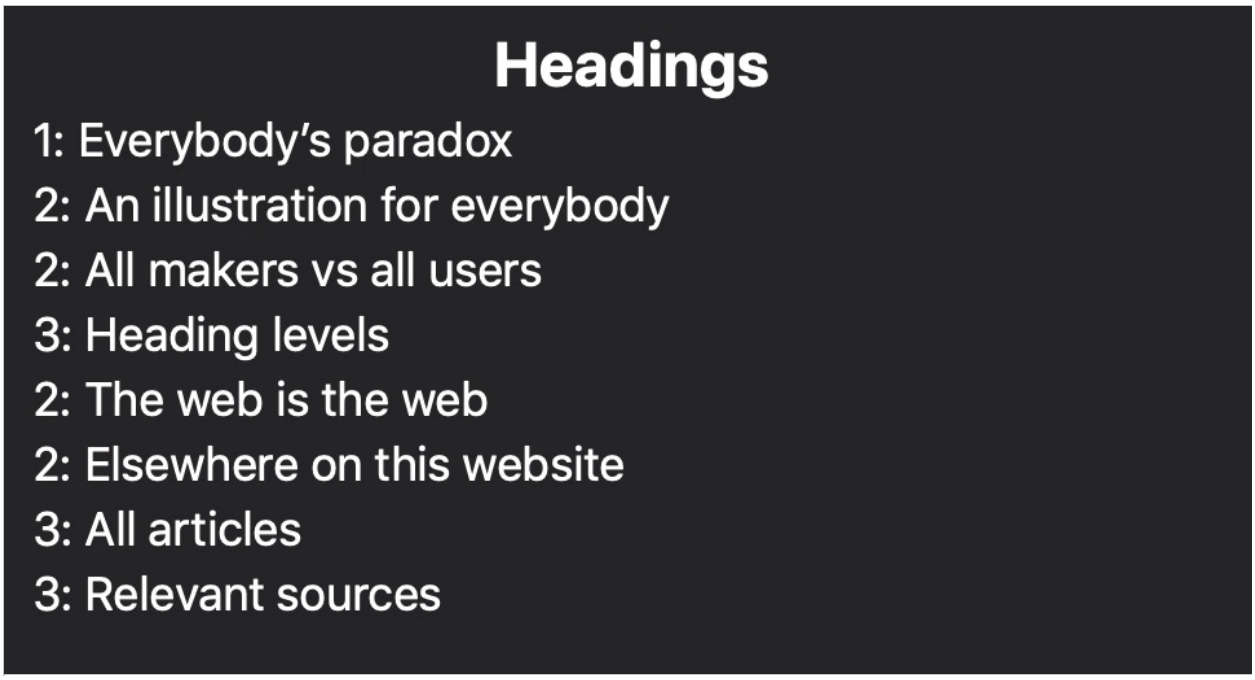
All makers vs all users

It would be true in practice as well, if everybody who creates things for the web was a web technology expert, a user interaction expert, and an inclusive designer at the same time. Creating things that really work for everybody is complicated — if possible at all — and needs expert knowledge.

If this expert knowledge was required in order to put anything on the web we would both make and break the web at the same time. On the one hand it would result in completely accessible interfaces, which would fix the web for people like Simon Dogger, who is blind. It would fix it from a user's perspective. But it would break the web for Simon as well, but this time from a maker's perspective: Since he is no web expert himself, he wouldn't be allowed to publish anything on it.

Heading levels

The use of heading levels is a good illustration of the conflict of interests between users and makers. Heading levels can help blind users in understanding the structure of a web page. They can ask their screen reader to turn these headings into something that resembles a table of contents of the page. But this feature only really works if heading levels are used properly.



The headings of this page, with their different levels. It gives a clear impression of the structure of the contents of this page.

Structuring heading levels is complicated. On many, if not most websites, document structure is not accurate at all.

Here’s a video of a screen reader showing all headings on an ill-structured page. It basically shows a seemingly endless, randomly numbered list.

View this video online: [vvg.gr/qz](vvg.gr/qz)

The random numbering, and the excessive use of headings turns this otherwise very useful feature into a useless exercise.

Misusing headings levels like this is common practice, even on larger websites. It stems from a lack of understanding. Which is possible because you are allowed to publish on the web even if you don’t understand the basics.

The web is the web

In the past there have been discussions about restricting who is allowed to publish on the web. For instance the discussion about certification for web professionals in 2007.[7] And more recently Mike Monteiro wrote a piece in which he argues that design should be a protected profession, just like architecture and law.[8]

I find what-if exercises about only allowing certified experts to publish on the web to be only mildly interesting. The conclusion would inevitably be that the web would cease to be the web. I am more interested in ways to make the web more accessible in its current form, without changing the principles of the web itself. The web is a chaotic, complicated medium and it only gets more chaotic and complicated because everybody is allowed to publish on it. I am not going to try and change that.

My research has focused on other directions of possible change. By promoting a more critical design *attitude* within the web community for instance. By having a critical look at the design tools and some assistive technology. By using an inverted, more focused approach to inclusivity. And by having fun.

In the next chapter I will talk about the *weirdness* of some of the defaults we have to work with on the web.

The defaults suck

The web is a strange medium. One of its founding principles turns out to be an impossible paradox. So even in theory it is probably not possible to create a web that is truly for everyone. When we take a close look at the practice of the web — browsers, screen readers, design attitude — then there are a few more factors that don't seem to help either.

Focus

Take the default styling that different browsers use for their focus states. In some browsers they look like a dotted border, in others they look like a blue glowing border. Many designers think these states look ugly. So they want them removed. Focus styles were added to browser so people could browse websites using their keyboard. The practice of removing focus states is so widely spread that nowadays it's impossible to navigate the web using a keyboard alone.

But even if the focus states are not touched their default behaviour is not very well implemented. Here's a video of someone trying to tab through a list of links using the Firefox web browser.

View this video online: vvg.gr/ra

As you may see, when they reach the bottom of the screen the text of the link disappears behind the so called *status bar*. This is just one example of the broken state of default focus behaviour. There are many more. It shows that keyboard navigation is neglected by the people who build browsers as well.

When I asked Marijn Meijles why he never uses his tab key to navigate interactive elements, he explained that in part it's details like these that make keyboard focus too unreliable to depend on. Marijn taught himself a different

way to control his computer.

Screen readers

I've always assumed that adding detailed structural semantics to an HTML document is a good idea. I started to doubt if this is such a good idea when I observed casual computer users who depend on a screen reader. All elements that have some relevant semantic value, like heading levels, navigation items, links and forms get this meaning attached to it, which is spoken out loud. So this page would't simply start with the title spoken out, but it would sound like *heading level one, The defaults suck.* When a page consist of many elements, things can become very annoying.

An artist impression of a website that's shouting semantics at the user

My impulsive reaction was to create websites without any structural semantics in them. Bram Duvigneau pointed out that, while this might indeed help some casual screen reader users, it would very much cripple the experience for experienced users.[9]

So simply ignoring all semantics is not the solution. This problem is much more complicated. It is both a design issue[10], and an issue with screen readers

themselves[11].

Design attitude: checking off lists

Then there's an issue with the way many companies approach accessibility: not as one of the starting points of the design process, but as a checklist that's being ticked off after the product is released.

You could argue that this is still better than ignoring accessibility completely. But I prefer Kat Holmes's conclusion on this attitude:[12]

> *Treating accessibility and inclusion as an afterthought, or only meeting the minimum legal criteria, is an* exclusion habit
>
> *—Kat Holmes, Mismatch*

She argues that ticking off lists is a problem that we should try to solve. And I agree. Our ambitions should be much higher when it comes to designing for an accessible web.

This thesis is not going to solve these issues. There is not a copy-pasteable solution for the issues with tabbing in it. There is no practical fix for the issues with screen readers. And I am not going to present you the definitive solution for the *exclusive habit* of ticking off a checklist. But this thesis may give individual designers, or design teams a few tools and insights to work with. And in that case, this thesis may help a little bit in creating a more inclusive web.

So the web is weird, and our design practice doesn't help either. In the next chapter I will explain how I found my *Exclusive Design Principles*, the principles I used to create tailor made accessible interfaces throughout my research. The *Exclusive Design Principles* that gave the name to this thesis.

Flipping things

In the summer of 2017 I asked Léonie Watson to give a guest lecture at the University of Applied Sciences in Amsterdam where I work. I asked her if she could talk about *pleasurable user interfaces* for people who happen to be blind, like herself. I hoped that if the visitors of this lecture would see interfaces that are nice to use with a screen reader, they would be inspired to start designing similar interfaces themselves.

Unfortunately Léonie couldn't answer the question. To her an interface is a pleasure to use when she's able to fulfil the task at hand without any help. And even then her demands are low: if necessary she can fulfil the task *by using the developer tools to change the technical workings of the website*. From an interface design perspective these standards are unacceptably low, yet for Léonie this is the highest she can imagine.

There is a huge gap between the level of knowledge about designing interfaces for ourselves and designing interfaces for people with special needs. With this knowledge gap in mind I looked at a set of inclusive design principles.

Inclusive Design Principles

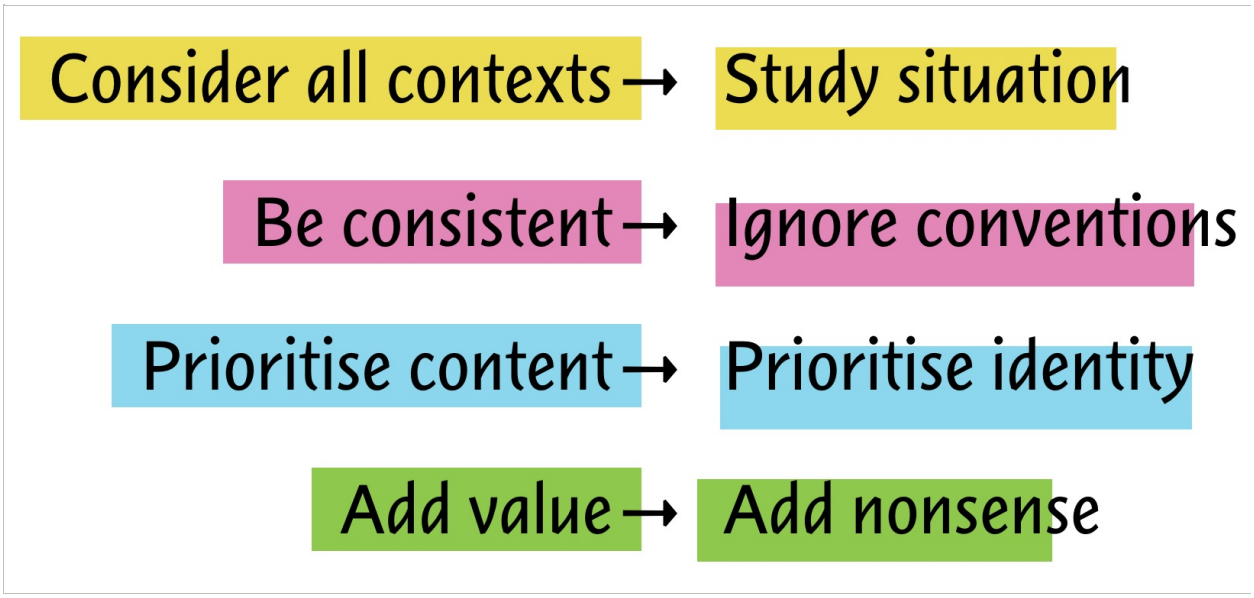
In 2017 a group of experienced web accessibility experts published a set of inclusive design principles.[13] Léonie Watson, one of the people behind the principles, explained why they made these principles in an email:

> *We created them because we think there is more to accessibility than technical conformance, and wanted to express that in a way other people could use for themselves.*

The idea is that if you use them, your websites will be more accessible. This set of principles was, in a way, the base of my research. But instead of using them as they are, I flipped them. At first I flipped them to test them. As Alla Kholmatova explains in her talk about design principles:[14] If you can imagine that someone else might be using the flipped versions, your principle is a good principle. In other words, the opposite has to make sense as well.

Another thing she explained is that a good set of principles has between three and five items in it. This makes it easier to remember them. There are seven principles in the original set, so I combined a few of them, and turned

them into four.



The four original principles, with their flipped counterparts

These are the four principles I flipped to see what happens.

Consider all contexts → Study situation

If you want to create an inclusive website it is important to understand all the different contexts in which people will use it. For instance, you have to consider people's abilities — do they have fine motor control, do they have eye sight? — you have to consider their hardware — do they use a large of a small device — and the software they use — is their browser up-to-date or ancient, and do they maybe use assistive technology like a screen reader. It could be argued that there are endless amounts of contexts to consider.

I agree that it is a good idea to consider all contexts, but only if the design team really understands these contexts. A lot of knowledge in the field of web accessibility is based on *hearsay*. It's knowledge gained from reading blog posts and books, and knowledge copied and pasted from code examples. Usability tests are not very common in the web world, let alone usability tests with real people with real disabilities.

One way to gain expert knowledge is by focusing on one single context for a while. And so the principle *Consider all contexts* turned into the first exclusive design principle: *Study situation.* That's what I did during this research. Together with a group of students we observed how Marijn, a visual keyboard

user, controls his computer. And I studied Simon Dogger and Hannes Wallrafen, both screen reader users, while they tried to use different websites. Which leads to the next principle:

Be consistent → Ignore conventions

The second principle says that you should *Use familiar conventions and apply them consistently*. If you have a convention that works very well in all different contexts, you should use it consistently. This is of course a very good idea. Inconsistency can confuse people.

This principle assumes that we have familiar conventions to work with. Again from observations I did, and from conversations I had with accessibility experts, I have to conclude that for certain users there is no such thing as a familiar convention.[15] Patterns we take for granted, like a navigation at the top of every page, make no sense to certain screen reader users.

So in order to come up with better patterns we should not be afraid to ignore the current conventions, especially if studying different situations leads us to the conclusion that the conventions might not be as convenient as we assume.

Prioritise content → Prioritise identity

The third principle tells us to *"help users focus on core tasks, features, and information by prioritising them within the content and layout."* This sounds almost too obvious to be a real principle. And flipping it seems to support this: The opposite of prioritising content would be something like *neglecting content*. There are certainly websites that could use more attention when it comes to their content, but I find it hard to imagine that someone would consciously use *neglect content* as a design principle.

But there's another way to look at this principle. Apart from good content, there are other factors that define the quality of a website. For instance *identity* plays an important role as well. Identity is interesting. There's brand identity, there's the identity of the design team, but there's also the identity of the people who use your website. We use these identities all the time. But what happens when we use the identity of people who have been excluded from the digital world? For instance, what if we use the identity of Hannes Wallrafen, a blind photographer I worked with? What if we use his worldview, and his *webview* as input for our designs?

Add value → Add nonsense

The last principle is hard to flip as well. It says to *"consider the value of features*

and how they improve the experience for different users." It doesn't seem to make any sense to *not* do this.

In the first two principles I concluded that we don't understand the different contexts, that we haven't studied the different situations enough yet, and that because of that we cannot assume that the conventions we find familiar are familiar for everybody.

So before we can add value for different users, we have to research different ways of adding value. One good way of exploring the possibilities is by allowing the design team to come up with ideas that may seem nonsensical. This is a rather common practice in early phases of design sprints. Ideas that sound ridiculous to some, might make sense in someone else's context.

I have used this principles in a few projects now, as I describe in the chapter about adding nonsense, and these nonsensical ideas sometimes turned out to be very valuable.

The idea of adding nonsense is of course not something I invented myself. For instance, it is also the whole idea behind the Ig Nobel Price. Ig Nobel Prizes are awarded to *"honor [scientific] achievements that first make people laugh, and then make them think."* For instance, in 2018 the *Economics Prize* was given to a team that investigated *"whether it is effective for employees to use Voodoo dolls to retaliate against abusive bosses."* And the 2018 *Anthropology Prize* was awarded *"for collecting evidence, in a zoo, that chimpanzees imitate humans about as often, and about as well, as humans imitate chimpanzees."*[16]

The Ig Nobel Prices show that apart from being useful, adding nonsense can be fun as well, which is a good reason in itself.

In summary

A short summary of this chapter is in place, since it is the basis for the projects I've worked on:

- The Inclusive Design Principles assume we have expert knowledge of how to design for excluded people.
- The Inclusive Design Principles also assume the patterns we use nowadays are well tested and good to use.
- By solely focusing on the content they ignore the possible insights that the identity and personality of the excluded people have to offer
- The principle of *adding value* makes sense, but if we accept the three points above we can assume that we haven't investigated all options yet. Adding nonsense to the mix can help in coming up with completely new ideas.

I asked a group of students to use the Exclusive Design Principles while designing an interface for Marijn. We started with assumptions, based on existing knowledge about keyboard interaction. And of course these assumptions turned out to be false.

Fuckup's mama

Last year I asked a group of students to design and build *a pleasurable user interface* for Marijn Meijles. Marijn is a computer programmer who has difficulty with fine motor control. He prefers to use his keyboard instead of his mouse to control his computer. The first prototypes my students made were based on the assumption that keyboard navigation is primarily done with the `Tab`-key.

Here's a video of a fancy looking interface that's being navigated with the `tab` key.

View this video online: vvg.gr/rb

After one week Marijn came over to our university to test the prototypes. When we tested the first iterations of these beautiful, `Tab`-key optimised interfaces with Marijn it turned out he never uses the `Tab`-key. Instead he uses a combination of the arrow keys, his space bar, the `Enter` key and his trackpad.

Here's a video of Marijn not seeing any of the fancy `tab`-key interactions.

View this video online: vvg.gr/rb

The new insights about how Marijn uses his computer were gained after one

week. This gave my students enough time to create well considered, and tested working prototypes of user interfaces that were easy to use for Marijn.

Here's a video of an interface that's been optimised specifically for how Marijn uses his keyboard. [You can test it for yourself by using your arrow keys on this website].

View this video online: [vvg.gr/rb]

This research didn't invalidate the assumption that keyboard users don't use their tab key. What it did show is that keyboard usage can be much more complicated than that.

Some of the solutions my students created were variations of *spatial navigation*, a type of navigation often seen in environments like TV-menus, tailored to Marijn's situation. Other solutions focused on the reason why he uses his keyboard in a specific way. This is described in more detail in the chapter about [adding nonsense].

This case nicely illustrates the first Exclusive Design principle which says that we should study situation. The assumptions we had about keyboard usage turned out to be insufficient to create an interface that works for Marijn. We really needed to observe how he uses his computer in order to come up with something that works.

More assumptions

The over-simplistic idea of how people use their keyboard was not the only assumption that proved to be false, or half-true.

Semantics

It is good practice to use *semantic HTML* on the web. One of the reasons why you should is that semantic meaning makes it easier for blind people to understand the function of elements, and the structure of a webpage. I have been teaching this to my students for years, and I've always done my best to use

proper semantics in the websites I built. One of the more shocking observations during my research was that for certain screen reader users certain semantics are more confusing and distracting than helpful. Structural semantics, like heading levels and website navigation, are spoken out loud by screen readers. Instead of helping, this extra information makes things more complicated when a user doesn't understand what a heading level or a navigation is for. With this idea I created a prototype of a website for Simon Dogger without using any structural semantics. Again, I got these new insights by closely *studying situation*, which in this case consisted of observing how Simon Dogger and Hannes Wallrafen use their computer. More details on this in the chapter about designing like it's 1999.

Transcripts

One of the first reactions on Twitter to the very first episode of my podcast was *"very nice, but I am deaf."*[17] Blushing with shame I published a transcript a week later. All episodes have been published as audio and a transcript ever since.

I assumed that all deaf people who were interested in my topic would be happy now. Until I saw a presentation by Marie van Driessche.[18] She explained that sign language uses its own grammar and its own structural logic, which makes it very hard for people who were born deaf to understand written texts. And when this text is a literal transcript of spoken language, it's even harder.

This is one of those issues that are hard to solve. The idea behind publishing a literal transcript is that it resembles the original most accurately. Other types of transcript, where you transform spoken language into proper written language, will always require some sort of interpretation. On the other hand, these types of transcripts *could* be more accessible to some people. A possible tool to solve these kinds of issues might be the *priority of constituencies* that I explain in the chapter about stress cases. Another option would be to use an official sign language transcript instead of a written one.

Again, this is an illustration of the fact that assumptions alone are not enough. You need to study situation, in this case how Deaf people may perceive your transcript. It's also a nice illustration of the third principle, which says that you should *prioritise identity*. In this case, actively using the identity of people who are Deaf, actively designing *with* them, might result in better inclusive interfaces for things like podcasts.

Assumptions about blind people

There are all kinds of assumptions we have about blind people. I remember the first time I met Léonie Watson during dinner, the day before a conference in Zurich. I told her about a script I had written which translates colour codes into spoken language. For me this was an entertaining conceptual exercise, I didn't really think of it as a real solution, because, as I asked Léonie, *"what use is colour to* the blind*?"* To which Léonie answered that there's no such thing as *the blind*. There are people who were born completely blind, for whom colour may be a hard to understand concept. There are people who are partially blind, who may be able to see some colour. There are people like Léonie who became blind at a later age who still have a *memory of colour*. To name just a few. So yes, this tool I created may very well be handy to many blind people. It might be handy for a colourblind art director I used to work with in the past as well.

In the past I've heard similar assumptions in brainstorm sessions with design teams. *"Blind people don't use our service"* is an almost logical assumption when you create a website with videos.[19] In these cases, instead of assuming, we should ask the people we assume things about. Again, this illustrates the first and the third principles, about studying situation, and about prioritising identity.

In the next chapter called *More death to more bullshit* I will talk about the weird assumption I had that all screen reader users are computer experts.

More death to more bullshit

Together with Simon Dogger we created an alternative version of the 2doc website. Simon is a product designer who happens to be blind. Simon is not an expert screen reader user. He understands some basics, but doesn't know how to use features like quickly navigating all headings, or directly jumping to the main content. Instead he has two ways to browse a website.

1. Sit back and let the screen reader read everything on the page from the top left to the bottom right, and start interacting when it seems like it has reached the thing he is looking for.
2. Actively skip past all links until he has reached the thing he is looking for.

Both ways are time consuming, and take up more time if there's more content on a page, like Bram Duvigneau demonstrated during his expert conversation.[20]

When we met for the first time I observed Simon trying to do some fairly simple, common online tasks like browsing an archive of online video documentaries, and buying groceries. Simon was unable to do any of these tasks by himself.

Death to bullshit

In 2013 Brad Frost gave a talk with the title *Death to Bullshit* at a Creative Mornings meetup in Pittsburgh.[21] In his own words, "Death to Bullshit *is a rallying cry to rid the world of bullshit and demand experiences that respect people and their time.*"

On the one hand this talk was born out of frustration. Frost was frustrated by the fact that websites were slow, they were hard to use, and that as a visitor he was often not respected, or downright deceived, and that every website was trying to scream as loud as possible for his attention with banners, pop-ups, newsletter subscribe thingies, etc.

But on the other hand it was triggered by a more practical reason. Frost had been a vocal promotor of responsive web design for a while, and he knew that a simple website is much easier to make responsive. And of course this line of thinking fits in an older engineering idea that says that you should always try to keep things simple.

Frost shows that our websites are *"superfluous, cluttered, clunky, or*

needlessly complex”, among other things. This makes things complicated for people to use. But all this complexity is amplified for people with disabilities.

More death to more bullshit

All the sites I tested with Simon Dogger had one issue in common. There was so much stuff on each page that it was very hard for Simon to find the material he was looking for. On the groceries website he had to listen to 25 links that have nothing to do with ordering groceries before he can start with the ordering process. Since Simon is not an expert screen reader user it took him a very long time to reach his goal.

When he opened the website with all the Dutch documentaries he was welcomed with the message that there are 80 headings and 150 links on that page. After a few tries he simply gave up, he couldn’t find the one link he was looking for that would give him an entry-point into the archive.

The same thing even happened on his banking site with one of the core tasks: He could not find the link to the page where he could transfer money.

Reconsidering our stuff

Almost all webpages start with a navigation with all kinds of links that try to convince you to go to another page. If you think about this for a minute it sounds ridiculous. Why would you want to start a webpage with pointing people in other directions when they just followed a link to this specific page?

Yet it is one of the most common patterns on the web: There is a bar with several navigation items right at the top of each page. For people who can see and use a mouse or a touch device it’s easy to simply ignore this navigation. But it’s exactly this navigation pattern that forced Simon to first listen to 25 links that tried to convince him to go elsewhere. For people like Simon it is a literal hurdle.

A navigation at the top is such a common pattern that nobody really thinks about it. It’s what everybody does, and it’s what we’ve been doing for as long as I can remember. Without any research it’s placed at the top of each new project to start with.

The navigation pattern could use some *exclusive design workshops*. It shows that simply following conventions is not always a good idea. It clearly illustrates that the current conventions are not the right starting point. It also shows there's a lack of expertise when it comes to designing for people with disabilities. If we had been working with people like Simon in our design research in the past, we would have come up with other, less obtrusive navigation patterns.

Let's work with people like Simon.

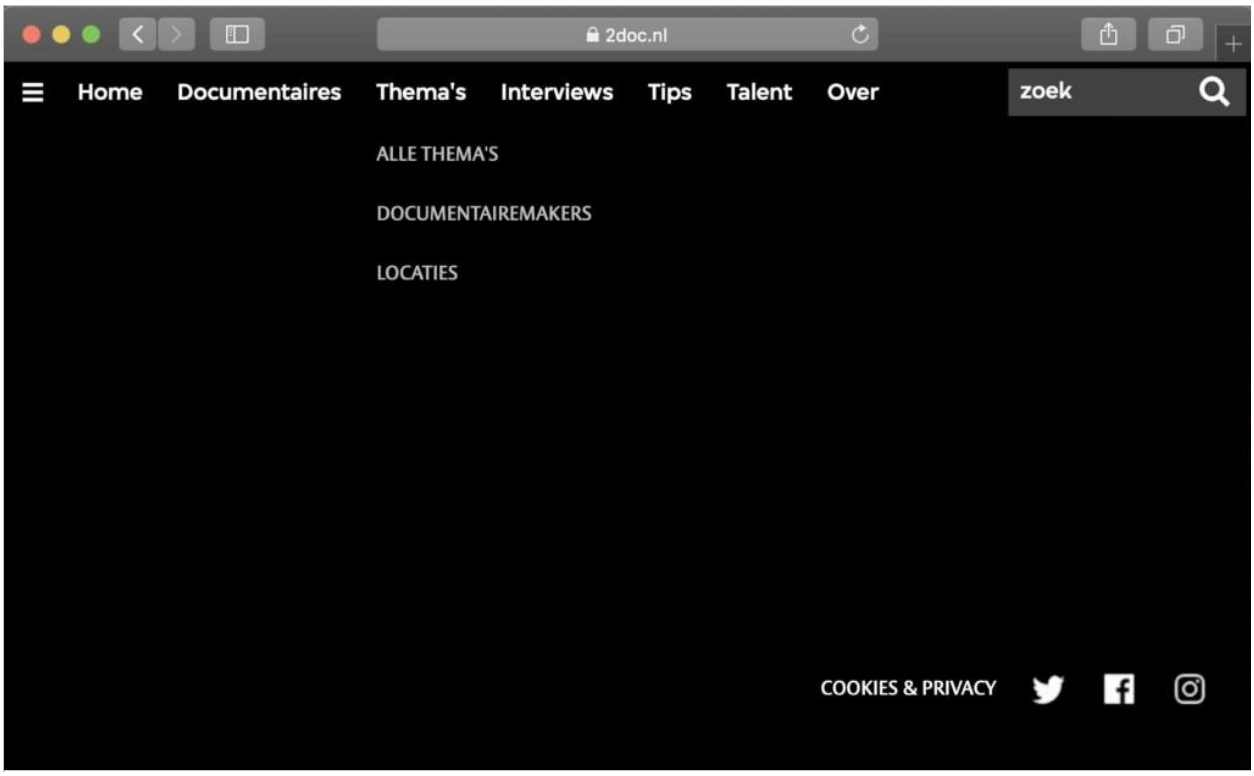
So what happens when you remove all the bullshit? In the next chapter I tell about the prototype of the 2doc website I made together with Simon.

Design like it’s 1999

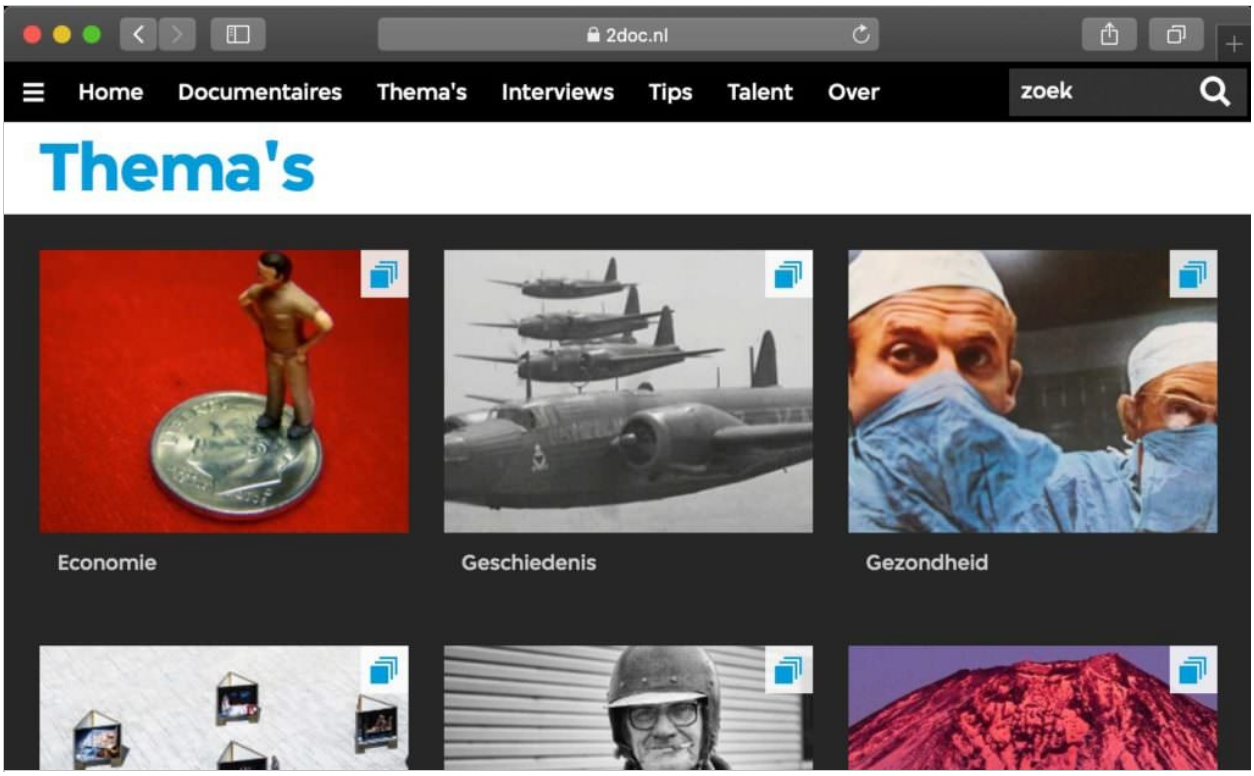
Simon Dogger is a product designer. He designs all kinds of physical products. Simon is not really a computer guy. Like most people probably, if he doesn’t have to use his computer, he doesn’t use it. One of the things he wants to do with his computer every now and then is to use the online archive of Dutch documentaries, 2Doc, both for fun and for his research.

When I visited the 2doc website for the first time with the intention to browse the archive, I took a quick look at the site, and without too much delay I found a promising link in the navigation that I clicked on.

For instance, when we click on the option *Thema’s*, we are presented with a drop down menu with the options to browse the archive via themes, makers and locations.

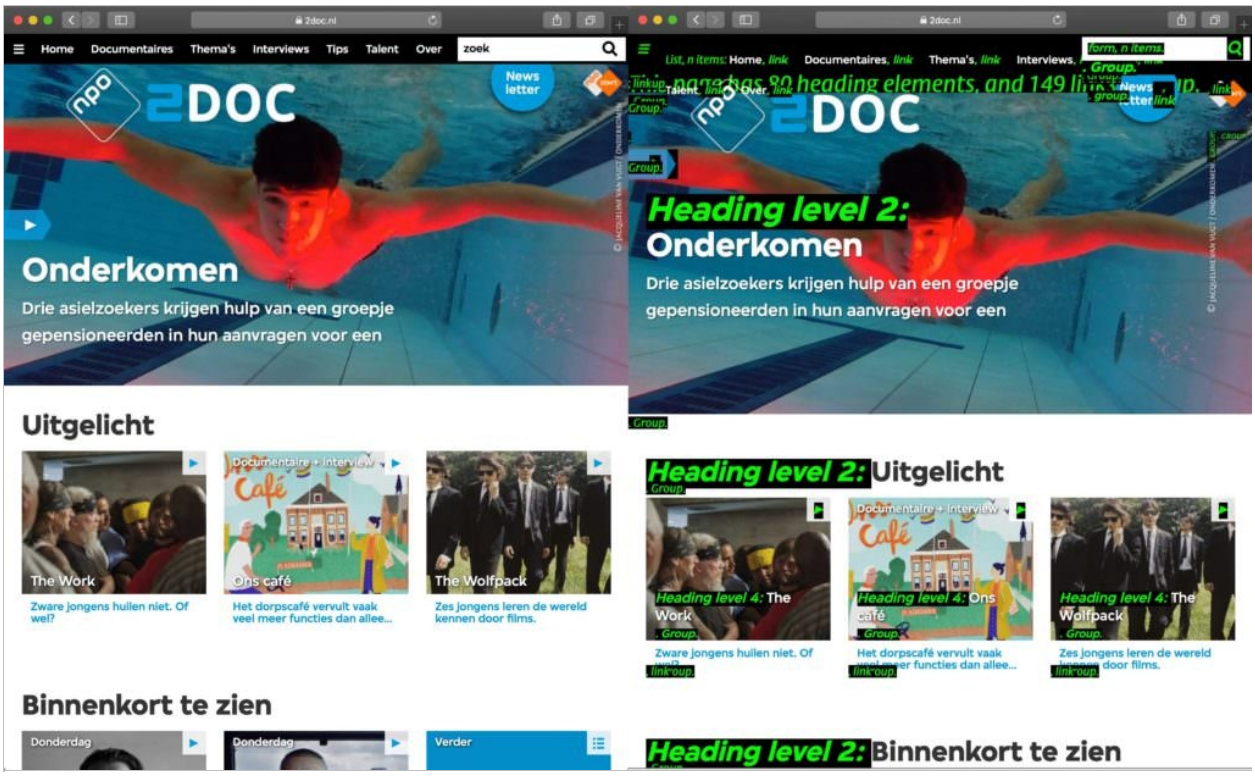


And when we pick themes, we are presented with this nice and easy page with a nice list of meaningful themes. From there on it’s pretty straightforward to find a specific documentary, or to be surprised with the various films that Dutch television has produced over the years. It is an incredible archive.



Back to Simon

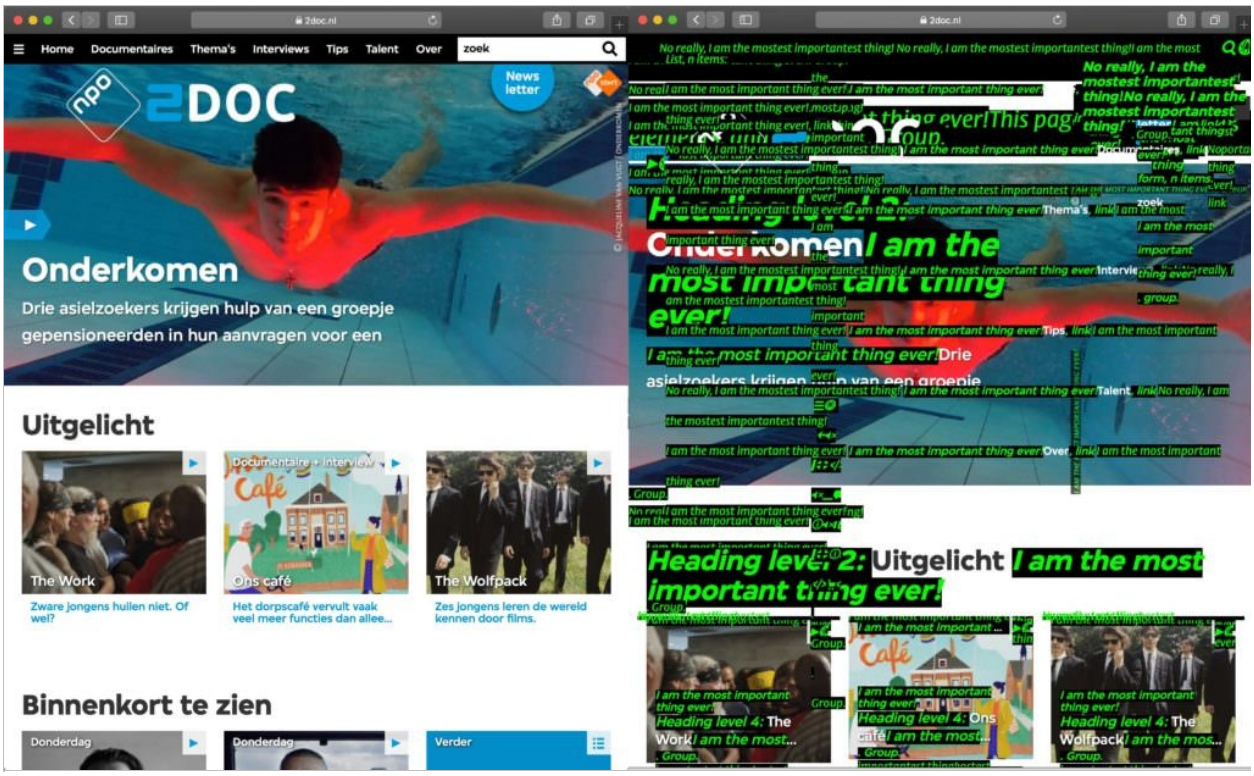
For Simon things are a bit more complicated. Simon is blind. When Simon visits this website, every single item on the page is spoken out loud to him by his screen reader. And not just the content of the item is spoken out, its meaning is stated clearly as well. For instance, a link to *Themes* will be pronounced as *themes, link.* And a heading for the section with new documentaries will be pronounced as *heading level 2, New documentaries*. I created a little tool to visualise this:[22]



As you can see, there's more content on the page for screen readers.[23]

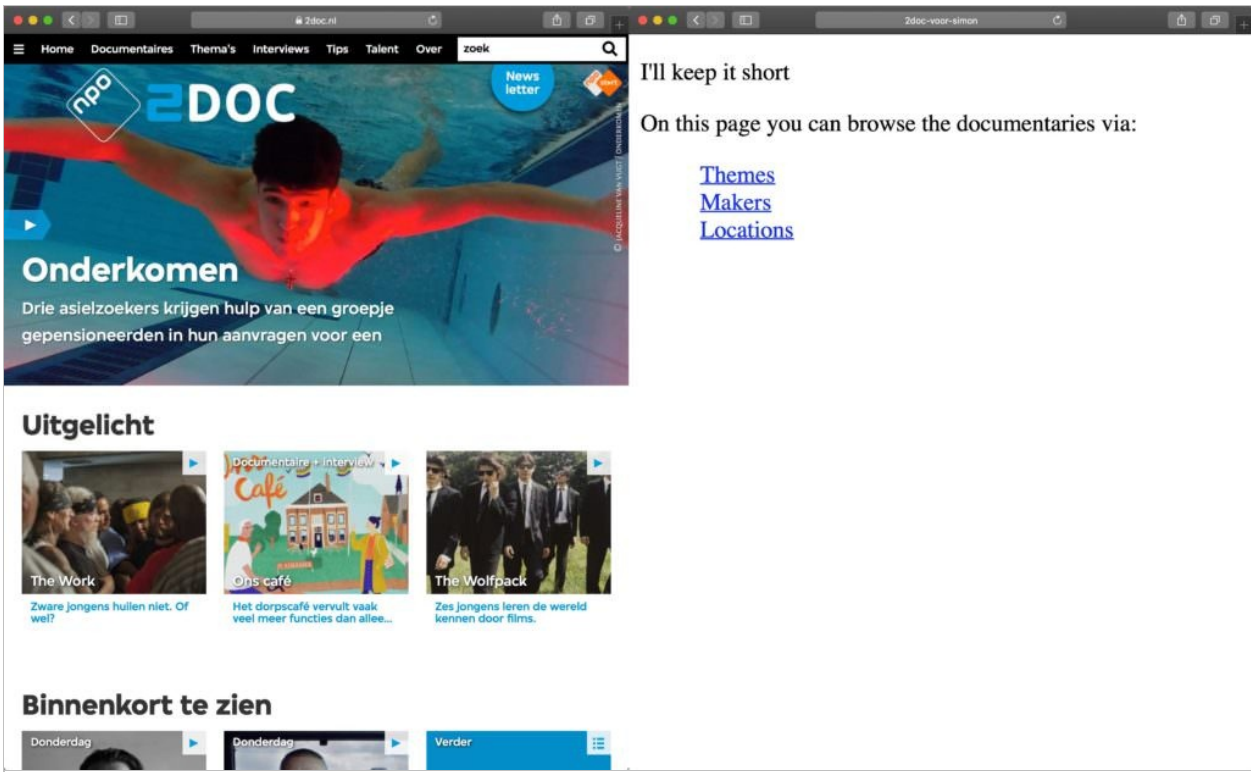
When the page loads the screen reader gives Simon some extra information about the page, so he can orientate a bit. It tells him how many headings there are (79) and how many links (149).

One hundred and fourty nine links. And one of them is the one Simon is looking for. I've watched him trying to find a link to the archives. The screenreader started reading elements, one by one. *Navigation! List! Six items! Home, link! Documentaries, link!* To him, every link, and every item on the page seems to be screaming: I am the most important thing on this page! With every single link he has to wonder if this is the one he's looking for, or if it is one of the other 148 links. I think the image where I show the extra content that screen readers add to the page doesn't visualise the audible experience well enough. I think this next image does a better job of showing the difference between the visible and the audible experience.



Simon quickly gave up. He simply cannot use this site. There are too many things on this page. And there are too many semantics he doesn't understand.

I made a redesign for him. It's pretty simple.



The homepage says:

> *I'll keep it short. On this page you can browse the documentaries via: Themes, link; makers, link; locations, link.*

That's it. This enables Simon to browse the archive and use it for his pleasure and his research. Simon was amazed. While he had tried many times before, he had never found these three thematic entrances to the 2Doc archive.

When I showed this design to Yuri Westplat, a colleague of mine and very experienced UX designer, he exclaimed: *"this is what websites looked like in the nineties! Where did we go wrong?!"*

Yuri seems to agree with me that there may be something wrong with our current conventions and that it might be a good idea to follow the second Exclusive Design Principle, which says that we should ignore conventions.

In this case ignoring conventions resulted in the very first website that's not a horrible experience for Simon. I'm not sure if this design is the ideal form for each and every person though. It could be there's a sweet spot somewhere between the extreme emptiness of my version, and the obese original with its 80 headings and 150 links. Coming up with a website that's fine for all stakeholders needs further research.

This is a nice example of how easy it is to make something usable for someone like Simon. But it doesn't really go beyond the usable. There is no personality in it, and we didn't really explore other possible solutions. In the next chapter I will tell about *the invisible animations* I made together with Hannes Wallrafen. In this project we went beyond the functional by working with both identity and nonsensical ideas.

Invisible Animations

During the summer holidays I read a book, written by Hannes Wallrafen, called *The Blind Photographer*.[24] In it Hannes describes both his life as a successful documentary photographer, and his life after he became blind, 15 years ago. What do you do as a photographer when all of a sudden you cannot see anymore? In the case of Hannes, he kept doing what he did before: he kept telling stories, but instead of photography he now uses sound.

Over the years Hannes has created all kinds of audio objects. In some of these he might explore a certain part of the city, like he did in the project *De stad bij nacht/De IJ-oeververbinding* which documents the nightly sounds of the ferries that travel from the north of Amsterdam to the center.

A screenshot of the current website with all kinds of unnecessary things.

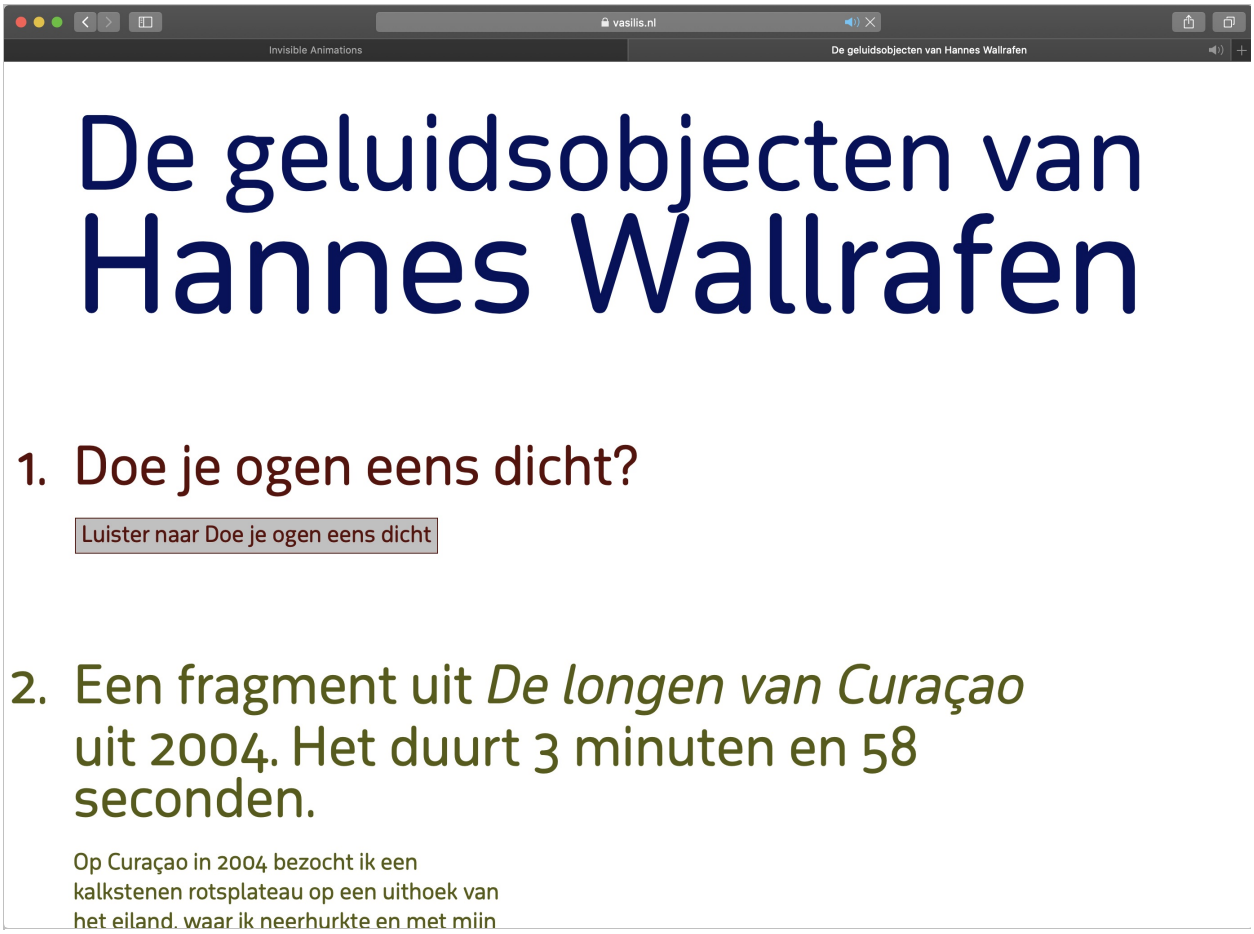
There's a small collection of these audio objects on his website. When I asked

him to show me this website on his own computer it struck me that the page itself is unnecessarily hard to use for him: the page starts with a long irrelevant navigation, the audio player is very hard to understand for someone who can't see it, and there are all kinds of non essential elements on the page that can — and thus do — cause confusion.

We decided that I would try and make a tailor made version of this page for Hannes.

It was pretty easy to get the page to a functional level: I removed everything that's not necessary, designed and created a new audio player that works nicely with his screen reader and a keyboard, and made it easier to skip from one audio object to the next.

Even adding a basic visual style was not too hard. Hannes has a clear preference for colour, he insists on high contrast and readability, and he is a very kind, very accessible person. These requirements combined with the subject resulted in a visual style that works. Hannes can't see it, but he did understand my explanation, and liked the sound of it.



A screenshot of the new, visual style of the webpage. Large type, colours, and

high contrast

While it is interesting to design a visual style for someone who cannot see, I wanted to take this design a step further. We had a few conversations about his personality, his character, and about how he socialises. This resulted in some improvements of the tone of voice, which is now more human, and not as systematic as before. As if Hannes is showing you around in person.

The really interesting step in the design process happened when I asked Hannes if could think of some nonsensical elements, some things that might not necessarily be needed, but could add to the experience. Hannes immediately knew what was missing on his web page: small, funny, illustrative animations. He wanted animations because they have a temporal dimension, which makes them perfect for illustrating audio, which is temporal and ever changing as well. But.

Animations for a blind photographer. What does that mean?

Animations are very visual. And I wanted to make this website first and foremost for Hannes. How could I make sure that these animations are funny for Hannes as well? Of course I could follow the current web content accessibility guidelines,[25] and create the visual animation first, and then try to create a textual description that comes close. But I have to agree with Kat Holmes who says that treating accessibility and inclusion as an afterthought is an *exclusion habit.*[26] So I decided to approach it differently: first I had to write the description, make sure it sounds funny with the voice of Hannes' screen reader, and only then make the animation itself.

Voices of screen readers nowadays sound convincingly human, yet very decent. Every webpage is read out with the same neutral emotion. From the voice you will not be able to discern what the page is about. Is it about some horrible disaster? Or is it an entertaining page with funny animations?

I tried to make Hannes' screen reader laugh, giggle, and sound surprised. For instance, one of the illustrations reads:

> *hè hè hè hè hè, so funny! All of a sudden there's a hammer that ticks the text down a bit.*

Another one:

> *Ha, look at that. Two ferries are ferrying from one side of the screen to the other, bouncing, boing boing boing boing, against the sides of the browser.*

If Hannes sniggers it's good.[27]

I haven't found any cross platform documentation of strings that give emotion to the different voices of screen readers. There is a large page that documents all special codes you can use for Acapela Group voices. They call these *vocal smileys*. Unfortunately this is not a standard used by other voice manufacturers. A platform independent overview of the different strings that make different voices laugh would be very welcome for people who want to create more emotional designs for screen reader users.

I started working on a simple application that tries to provide such a documentation.

Together with Hannes Wallrafen we managed to go well beyond the functional in this project. We did this by focusing on the needs of one single person, and by exploring nonsensical ideas.

Here I tried to design for the screen reader that Hannes uses. Which is hard. In the next chapter I explain that this has to do with the default settings screen readers use.

Semantics schmemantics

In the train back from Eindhoven to Amsterdam, after meeting Simon Dogger for the first time, I thought about giving up the web completely. We've been doing it wrong all those years, I thought. It cannot be fixed, was my conclusion. This was quite a depressing moment. What had happened?

Just before Simon had demonstrated how the web sounds to him. He had tried to do some common web tasks, like ordering groceries, transferring money and watching a video. And he had failed miserably at all of them. In part this was due to the large amounts of superfluous content that's on all the websites that he visited. And in part it's due to the way his screen reader explains all this content.

Semantic HTML

Screen readers don't just read the visible content on a web page out loud to the user, they also explain the *the meaning* of the content. So when something is a link or a form element the screen reader will say so, so the user knows that they can interact with the element. But not only interactive elements are explained. Things that explain the structure of a page, like heading levels, navigation and lists, are explained as well. The title of this page will be pronounced as *Heading level 1: Semantics Schmenantics.* Léonie Watson does an excellent job at explaining why this is such a powerful feature:[28]

> *HTML semantics are therefore important in two ways: We get a consistent understanding of content structure and native behaviour, and we get a common understanding of the content's meaning and purpose. The best thing of all, is that we get those things for free whenever we use HTML as intended.*

This is exactly what I had always assumed. Semantics can give a screen reader user an understanding of the content structure. Observing Simon left me in confusion at first. To him, hearing the semantics of elements, did not help him understand the meaning and purpose of the page. At all. The only thing it did was adding even more noise to an already cluttered page. Instead of helping him, it only confused and irritated him. *"What is a navigation and why does every page start with it, instead of with the content I expect to find?" "What does all this* heading level 5, 2, 4, 3 *mean and why does everybody put that in their*

pages?”

Expert users vs laypeople

The main reason why Simon doesn’t understand semantics is simple. Like most people on this planet Simon is not a web content expert. Words like *navigation,* or *heading level,* are simply not part of his vocabulary. Instead of making web pages easier to understand, hearing this jargon over and over again make them more complicated.

Do we expect everybody who uses the web to understand semantics on an expert level? No, of course we don’t. We make sure the semantics are visually clear. You don’t need to know that a navigation is called a navigation when you see one. And you don’t need to know that a heading is called a heading and that it’s of the second level. We simply see the hierarchy. It stands out because it’s styled like a heading. We don’t need to know the word in order to understand it.

How would you pronounce headings when you read out this page? You would probably pronounce them a little bit differently than a paragraph. Maybe you add a bit of emphasis. And maybe you add a pause before and after the heading. You could consider this to be aural styling of headings.

Screen readers lack the tooling to properly style headings.

What can we do about this?

I could have tried to teach Simon a thing or two about semantics and about using more features of his screen reader. I could have changed a few settings in his screen reader for him as well, in order to make it less verbose. This might have solved a few issues for Simon personally, but it would not solve anything for all the other people *like* Simon.

My first reaction when I saw that semantics confuse people like Simon was to stop using semantics. And when you look at the prototype I made for him you’ll see that it consists of only a few paragraphs, links and one single button. To Simon this was a relieve. Finally a website that’s not shouting incomprehensible words at him.

I discussed this idea of completely leaving out semantics with Léonie Watson and with Bram Duvigneau. To them, both expert screen reader users, these semantics really help in getting a better understanding of a webpage, and it helps them with navigating webpages easier. Not using semantics at all would completely break the web for them.

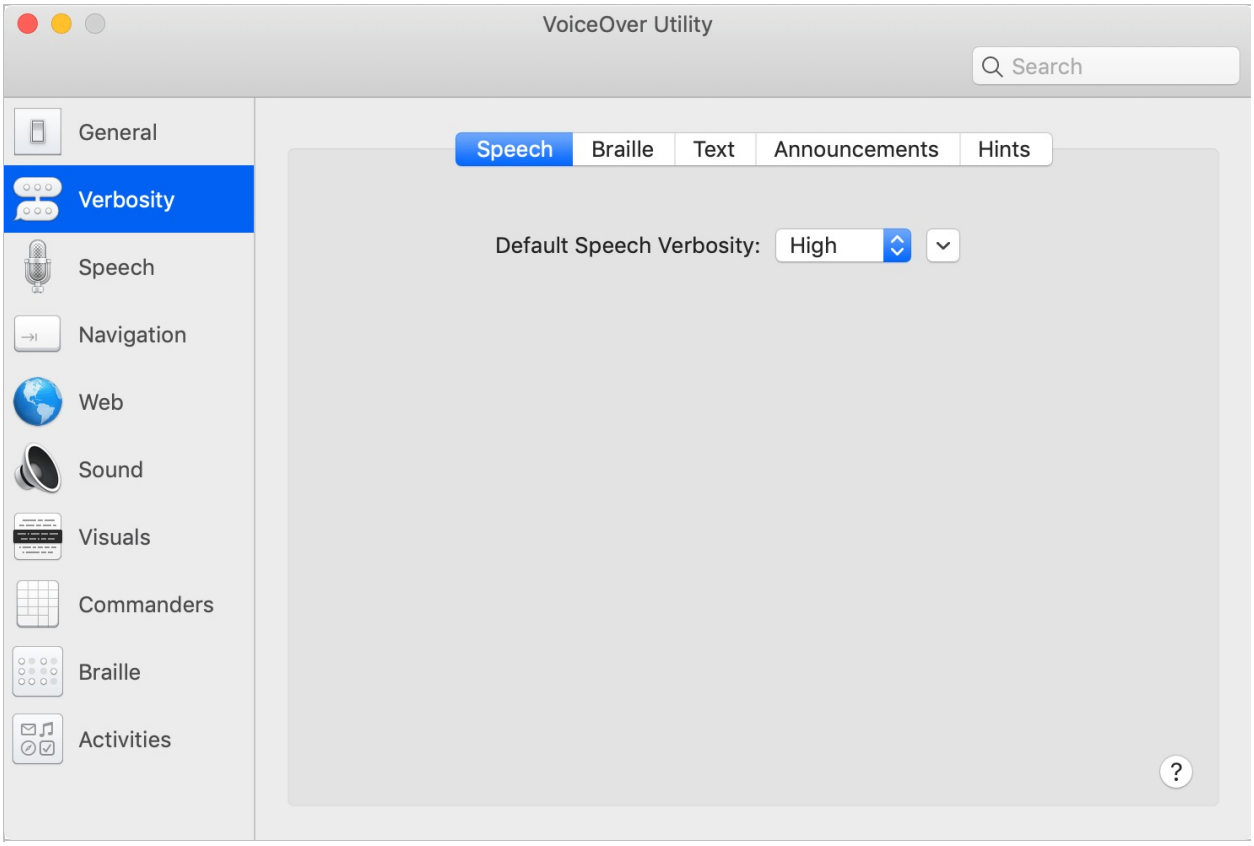
The solution is not in the way we write our HTML, it’s in the tooling.

We need better screen readers

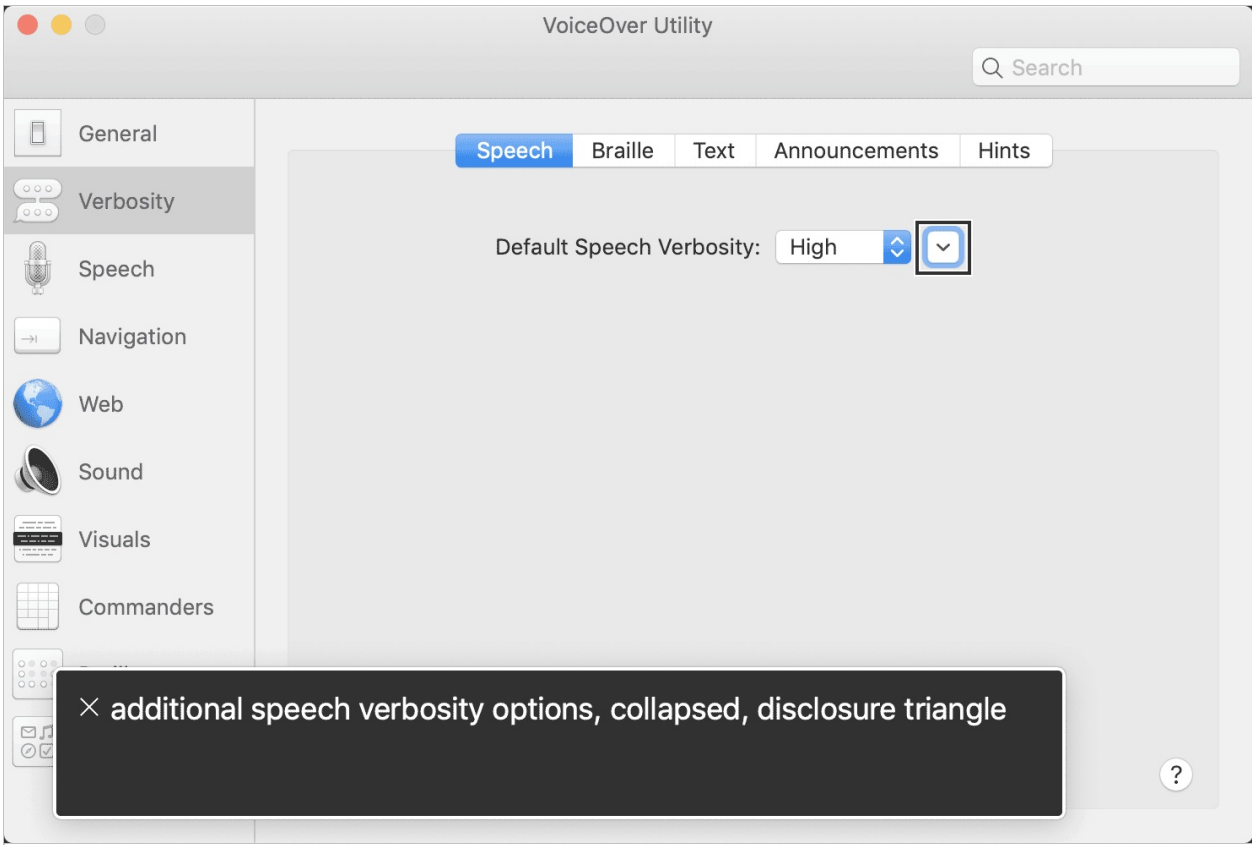
A quick fix would be to change the default settings of screen readers. By default, instead of all semantics, they should only speak out the behavioural semantics. Knowing that something is a link is essential, knowing that something is a heading is handy, but also needs expert knowledge that you can't expect laypeople to have.

It's true that you can change the level of verbosity in the settings of your screen reader. But software settings are not something that non-experts change that often. They are often quite complicated to use. They can be daunting.

If you want to tell VoiceOver on the Mac to stop reading out all headings you first would have to find and open VoiceOver Utility.



And then tab to the menu item Verbosity (and understand that this is the place to change the setting, and not Speech, Web, or Sound).



You will have to understand that you have to stay on the Speech panel, and then click on the button that's labeled Additional speech verbosity options, collapsed, disclosure triangle.

This will open a long table in which you can change the verbosity of heading levels. Which again is rather complicated. All in all, there are many steps, and many more possible wrong roads to take among the way. Now try to imagine what it would be like of you were trying to do all this without being able to see.

You cannot expect people who are no experts at using their computer and screen reader to understand how to change these settings.

One option would be to reconsider the way that changing settings work. Instead of a jargon filled maze, changing setting could be turned into a more user friendly interaction, like a conversational interface for instance.

Smart defaults

However you present them, I think that screen readers should reconsider their default settings. The default settings should be made for *normal* people who need a screen reader, and not for expert users. Normal people will get frustrated when software is needlessly complex, and give up using it. This is the opposite

of what we as inclusive designers want to achieve. Simon for instance only uses his computer if he really has to, because he absolutely hates the way it works. An expert user will get frustrated as well, but chances are higher that they know how to change some settings.

Changing the default settings may help in removing complexity for laypeople, it will not solve the problem of styling heading levels. If heading levels are not spoken out as such, then there will be no difference between a heading and a paragraph. This could be very confusing as well.

Next level screen readers

There is a lot more to improve when it comes to the user experience of screen readers. It would be interesting to see what would happen when UX design universities started experimenting with the open source screen reader NVDA. What happens when indeed you change the default settings like I suggest? Would it be possible to add a little bit of intelligence? If a screen reader were a bit opinionated it could for instance ignore annoying patterns like the navigation.

Maybe a next step for screen readers would be to listen to story tellers, and people who read books to children. How do they emphasise a next chapter? What are the stylistic details they use? How do they make sure hierarchy is clear. Would it be possible to add more emotion to screen reader voices? Could you for instance translate the visual style of a web site, into something like an *audible tone of voice*? And could we please create a standard way of making screen readers laugh?

> So screen readers should reconsider their settings. And it’s clear there is much room for innovation in this field. In the next chapter I explain that a change in design attitude could help as well.

Stress cases

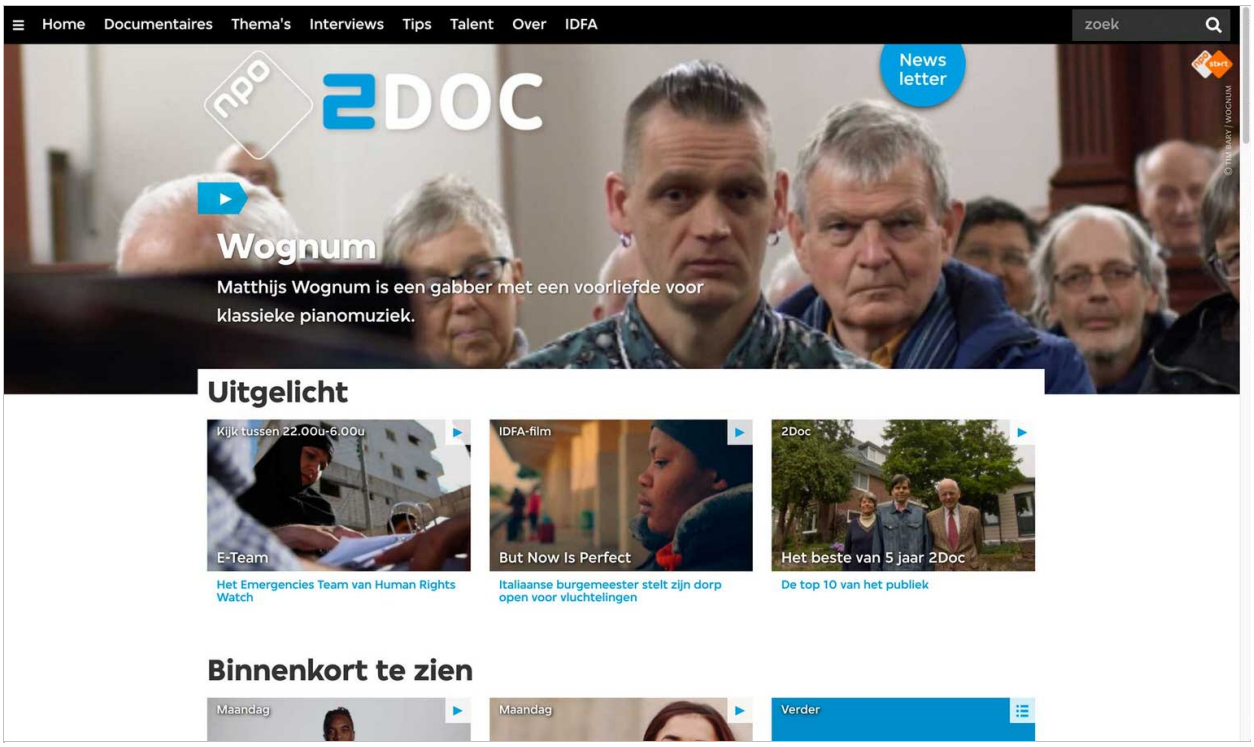
The HTML5 specification uses a wonderful set of rules to decide who is correct when there's a conflict of interest between different stakeholders. This document is for people who come up with new features for the web. This is the hierarchy they use.

1. End user
2. Designer
3. Browser maker
4. Person who specifies the feature
5. Theoretical purity

This so called *priority of constituencies*[29] can be used on a smaller level as well. When we look at the end user only — the one who is always right according to the HTML5 spec. That's not a single person. There's a wide range of users who sometimes seem to have conflicting interests as well.

Setting priorities

For instance, when you look at the different users who need to get a clear hierarchical overview of a webpage: who will have the most difficulty getting this overview? Let's make this a simple binary choice between sighted users and people who are blind.



The sighted user can immediately see all kinds of visual clues. Within a fraction of a second things like font size, white space, colour, ratios, *gestalt*, help in understanding where an article starts, how long it is, and what else is on the page. This doesn't just help them in understanding the structure of the page, it makes it possible to simply ignore the things that don't matter, like banners, sidebars and navigations.

There are no visual clues for screen reader users. They have to use tools like reading out all headlines to get an idea of the table of contents of the page.

View this video online: vvg.gr/rc

This is a much more active method of getting a sense of hierarchy. Comparing this to the immediate sense of hierarchy a sighted user gets, we can agree that

it’s probably harder for the blind person to get a clear overview of the page. If we use this conclusion for our priority of constituencies, this would mean we should prioritise blind people in this case.

On the other hand, Léonie Watson pointed out to me in an email that this hierarchical overview that screen reader users get by listening to the heading level structure, is something that sighted users do not get.

> *Perhaps a different question is this: if no-one was ever able to see (if we were all blind), how would we think about accomplishing this goal? One approach we use in print is a table of contents. I don’t actually recommend this so much for websites (though it works for your thesis), but hopefully it illustrates one way in which we give people a holistic sense of a larger thing without actually being able to see it all?*

Possible priorities

There are all kinds of different users to consider, and in different situations the order of priorities changes. For instance, keyboard users are not always first priority: Filling in a form is easier for someone who use a keyboard on a desktop computer than for someone who uses a touch device. On a desktop computer there is much more space, so it’s easier to keep a good overview of the context of the form. This is much harder on a small device, so extra attention to this design is needed.

Here are a few possible priorities of constituencies for different types of patterns.

Navigating a webpage

Blind person → non blind

Typing something into a textfield

Touch user → keyboard user

Controlling an interactive element, like the date picker gantt chart.

Keyboard user → Mouse user

Blind user → visual user

A Podcast

Deaf person → non deaf

An illustrative animation

Blind person → non-blind

Reading a long article
Sighted user → screen reader user

Setting clear priorities of constituencies could be used as a tool for achieving what Kat Holmes calls the first step in inclusive design: *recognise exclusion.*

Stress case design

Sara Wachter-Boettcher and Eric Meyer call this type of design *stress case design*, which they use as an antidote of *edge case design*.[30] They explain that instead of focusing on the imaginary average persona, we should focus on people who will visit your website in stressful situations.

Ironically, working with edge-cases in software engineering is a method where the extremes are actively tested to see. What happens when someone fills in an extremely long name, or an incredibly high number? By testing these edge cases you make sure the application still works in extreme situations.

Somehow in the web design world the term *edge case design* turned into the exact opposite, and thus became an exclusion habit. Or in the more strong words of Wachter-Boettcher and Meyer:

> *[Design team:] "We're designing for the 90%, not the 10%." That's classic edge-case thinking: a shorter way of saying, "That's a difficult use case that I don't want to think about."*
>
> *—Design for Real Life, chapter 3, Incorporate Stress Cases*

Using a word like stress case instead of edge case may help. You don't want to let down the people who visit your website under stress. On the contrary, these are the people you want to help in the first place.

If we want to design for stress cases we need to take the second Exclusive Design Principle at heart: ignore conventions, stop copying, and start designing.

Coders should learn how to design

When I started my Master Design course I had a fellow student, an engineer, who is specialised in air circulation systems for large indoor sports complexes. She had worked on quite a few of them. She explained the process as follows: when a new sports complex is to be built she receives the numbers: numbers about visitors, numbers about budget, numbers about size. With these numbers she looks at current systems and comes up with a few options.

The reason she started this Master Design is because in all those years she *had never spoken to a single visitor of the buildings she had worked on.*

This reminds me of the web design practice.

Component based engineering

In recent years more and more websites are based on generic component libraries like Bootstrap and React. If you want a navigation, you simply add the navigation module to your project. If you need a form, you use the form the library offers you. If you need layout, you can use the layout module.

Websites nowadays consist in large part of these ready made components made by others. We simply copy and paste things that others created — for entirely different contexts — into our own contexts. In the description of his talk Stephen Hay says it like this:[31]

> *When thinking about how to solve our design problems, we often look to what others are doing[…]. We call this "inspiration". But at the heart of it is a focus on solutions rather than problems. We fall in love with a solution to someone else's problem and try to make it fit our own. We contribute to design sameness, and confuse it with reasoned convention. In making things easier for ourselves, we might miss opportunities to really make a difference.*

We really need to make a difference when it comes to web accessibility.

Design Meets Disability

In the summer of 2017 I was struggling with my research. Léonie Watson had just given her talk about *(the non-existence of) pleasurable user experiences for screen readers*. I knew I had to investigate *pleasantness* in some way. That summer I read the book *Design meets Disability* by Graham Pullin. In this book

Pullin argues that adding a layer of design to products makes them more human.

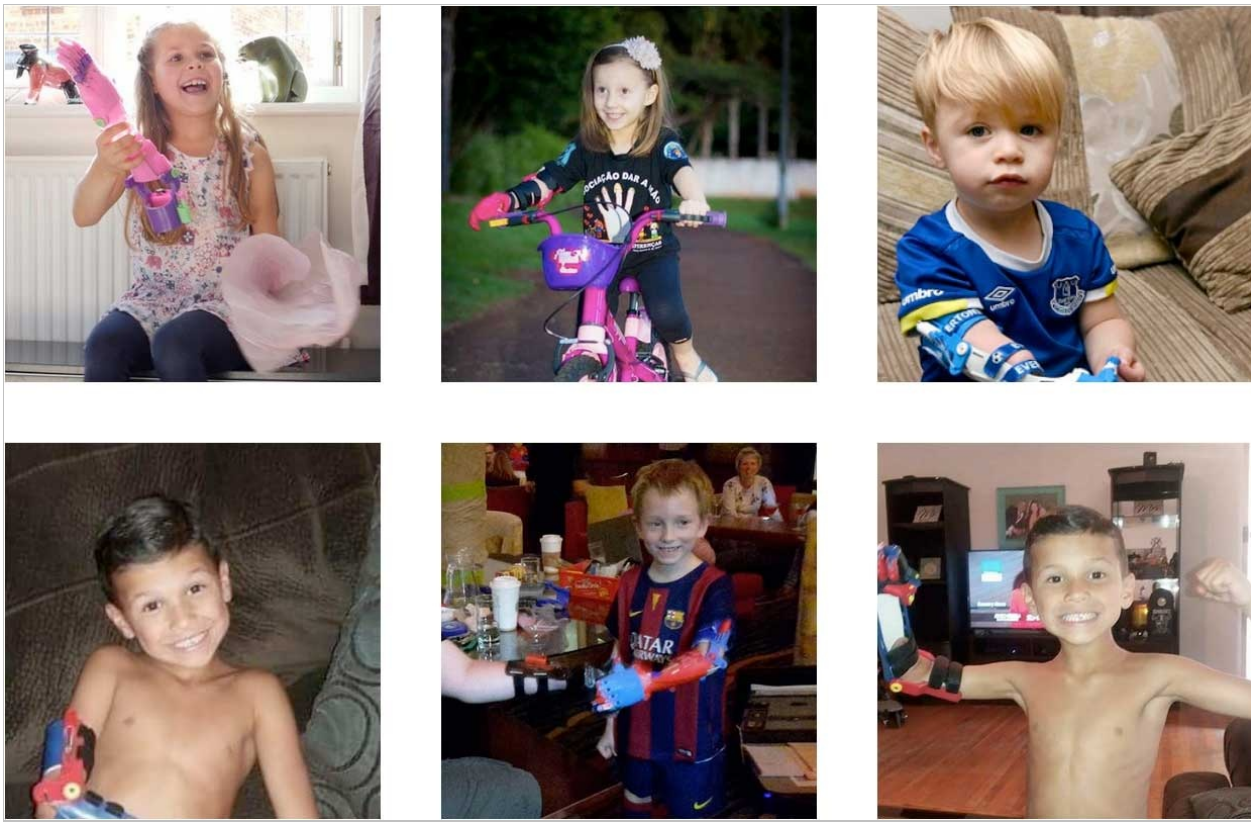
Pullin explains that current products for disabled people are mostly engineered: when they work, they're done. The focus is solely on the functional.[32] Things like emotion and personality are not considered. He gives the simple example of what happened when designers started designing glasses. 30 years ago, it was completely uncool to wear glasses. They were pieces of glass in iron wire. With a bit of luck you could choose the colour of wire. Kids got bullied if they had to wear glasses.

In the 90s designers started to treat glasses as fashion accessories. You can now buy a pair of glasses for every other occasion, matching glasses for each shirt. And all of a sudden it's cool to wear glasses. My kid cheered when she found out she needs them. Thirty years ago it would have been drama.

You could argue that this is wasteful luxurious design. Who needs more than one pair of glasses? But there are other examples. For instance, Pullin shows an example of different prosthetic legs, all for different occasions. It makes complete sense to wear different legs for running and other legs for partying. Again these are luxurious goods, tailor made and probably expensive, but they definitely lift the quality of life.

Prosthetic arms for kids

If you need more convincing of the power of design: This is the wonderful example of Team Unlimbited. They work from a shed in their backyard where they create cheap, yet tailor made prosthetic arms for kids. If these prosthetics had been purely functional it would have been wonderful, but what really makes this project stand out is the fact that the kids can choose their own colours. Now all of a sudden they are not simply functional things, they turn into something cool.



Of course you can have a Wonder Woman arm if you want one! © Team Unlimbited

Design can help if we want to lift digital experiences for people with disabilities to a next, more personal, more emotional level.

Next to the web accessibility *engineering* expertise we need more inclusive *design* expertise. Copying code examples and commonly used patterns will not answer complex questions like *how do you add emotion to screen readers? How do you make keyboard navigation pleasant? How do you translate the tone of voice from an audio podcast for someone who is deaf?*

These are examples of complicated design questions that can't be solved by engineering alone. A different design attitude can help. As Kat Holmes explains, it all starts with recognising exclusion, and actively working together with those who have been excluded.[33]

The exclusive design principles are a good way to start with this. Studying situation helps in recognising exclusion and understanding the needs of the excluded. Prioritising identity helps in actively involving excluded people into the design process. And ignoring convention will help if it turns out they don't work after all.

And then, as Pullin explains, we need to add a layer of creative thinking *and* doing on top of that. And finally, if it's up to me, we should add a layer of nonsense to the mix as well.

Add nonsense

Marijn is a software engineer. He can only use his left hand, and with that hand he has difficulty with fine motor control. This means typing can be hard for him. It's even one of the reasons why he didn't finish university. He had to write papers. Marijn is incredibly intelligent, so thinking up a paper is very easy. Yet typing it would take him ages.

Basically the problem he has with keyboard layouts is that not all keys are next to each other.

On a QWERTY keyboard it's easy for him to type the word was because the letters are close to each other.

Typing the word is is much harder though. He will have to move his hand and exactly point it to that one small square.

After observing Marijn using his computer, Rick Buter, a student of mine, explored alternative ways of typing words. What if, Rick thought, you would only need a few keys?

What if you could type complete sentences by only using the u, i, o, j, k, and l keys?

And so he created a prototype where with only six keys you can type common words.[34] On this video you can see me typing the words *exclusive design*.

View this video online: vvg.gr/rd

Allowing weird ideas

At first Rick didn't dare to mention his idea, let alone create a working prototype. When he first thought of it, it simply seemed too ridiculous. Only when I explicitly asked my students to come up with nonsensical ideas did he dare to make a prototype and test it.

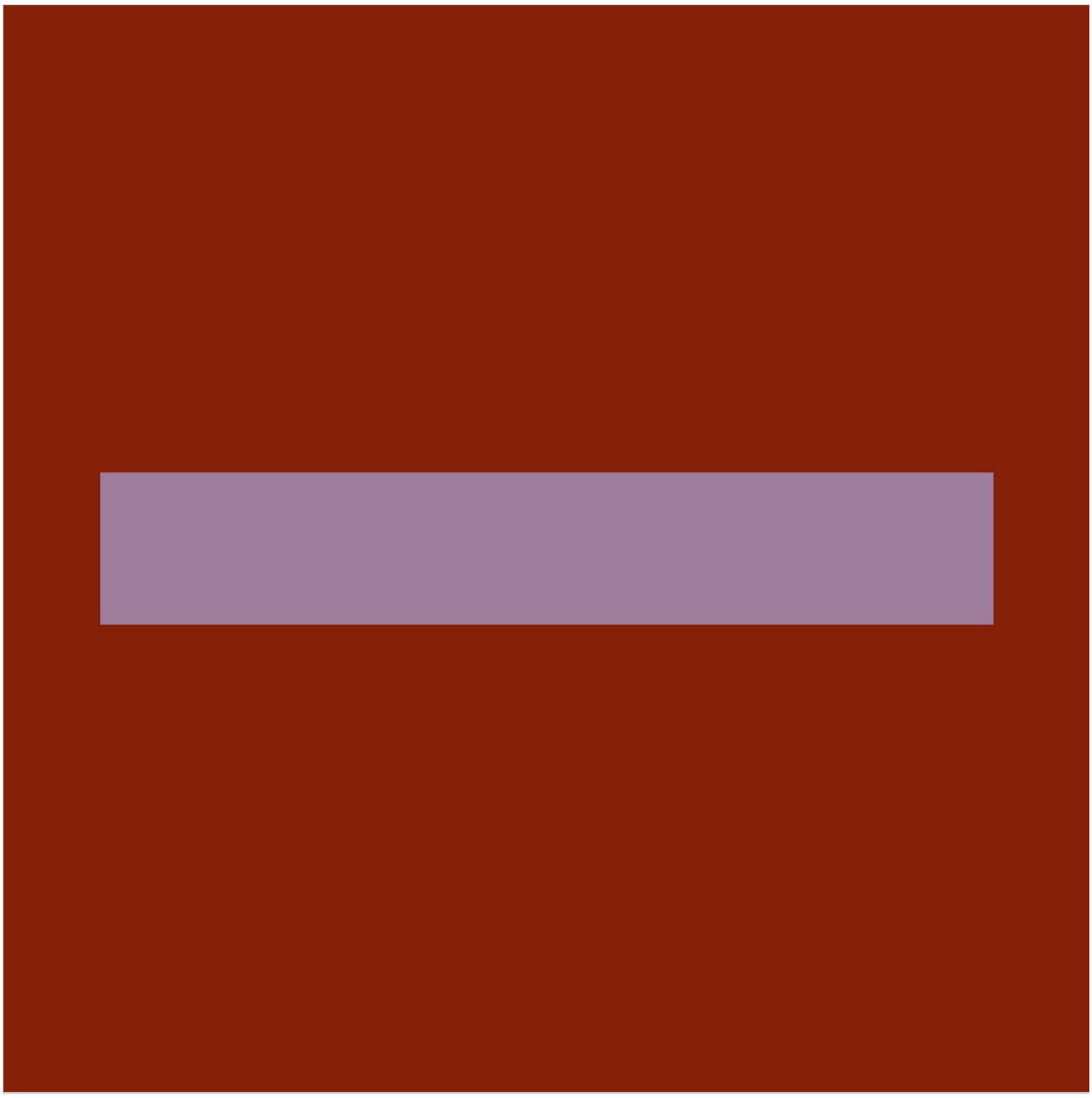
When he first showed this prototype to Marijn, he had to laugh indeed. It looked so ridiculous and complicated, it didn't seem to make any sense. Yet after trying it a few times he had to conclude that this might actually work. If you want to you can try the prototype for yourself.

Allowing ideas that seem nonsensical into a design proces can lead to innovative products. This is a common ideation method used in many different

creative fields. Questlove explains in his book about creativity that some musicians play music backwards in order to come up with completely new sounds[35] (not just to listen to hidden satanic messages).

My personal interest in nonsense has grown while I was working at a large web design agency. Everything we did there had to make sense, by which I mean that everything had to have an easily measurable effect. More visitors, more money, better performance, simple things like that. And thus we only investigated the obvious. This always gave me the feeling that we were missing out on things.

Back in 2010, as a reaction to this professional focus on *utility* I started a website called Love Nonsense, in which I blogged about nonsensical, mostly funny but often insightful topics. I have worked on several nonsensical side projects as well. Like this script that translates colour code into written language.[36]



A very unsaturated magenta rectangle — very wide and not high at all — on a rather saturated, dark red background

At first this idea of making colour accessible to blind people seemed nonsensical to me, since I assumed blind people can't see colour. Later it turned out that some blind people can see colour, and others may have memory of it, like I explained in the section about Assumptions about blind people.

Just like the six-key-typing project by Rick Buter, this is an example of using nonsense in order to allow weird ideas, and exploring them. And in these cases they resulted in valuable new insights.

Apart from a simple ideation tool to allow weird ideas, there are other reasons to use nonsense as well. Other reasons are lifting ideas beyond the obvious, breaking out of a single rusty context, exploring the unexpected, and of course having fun.

Talking dogs

An example of using nonsense to lift ideas beyond the obvious is this prototype of a talking dog. During my master research I organised a few Exclusive Design Challenges. Some of the more interesting ideas grew out of the *add nonsense* principle. Like this idea of smart glasses that give a blind person some information about the surroundings. This is a quite common idea, and working implementations of this already exists.[37] This team took this obvious idea to a next level by attaching the speaker to the guide dog.[38] All of a sudden all kinds of new possibilities and questions emerge. What's the personality of the dog? Will the dog be able to tell persons who want to pet it that you really shouldn't? Will it crack jokes? How will people react when they are being described by a talking dog? It would be so interesting, and so much fun, to explore this idea further.

An electric wheelchair-mouse

At another one of these workshops one of the teams had a lot of fun, together with Marijn, with working on an idea of using Marijn's complete electric wheelchair as a device to control a website. They envisioned Marijn moving the mouse cursor by driving through a large room, filled with sensors. While possibly spectacular, this of course is a rather unpractical idea. Yet it did contribute to my simple spatial navigation experiment,[39] which can be seen as a more economical and more practical version of this wheelchair mouse. This is a nice example of just having fun, which is important as well when working with a serious subject like accessibility.

Accessible video documentaries

A nice example of breaking out of a single, rusty context by allowing nonsense is the project I did with Simon Dogger. He asked me to create an accessible version of the 2doc website[40] for him, a website with video documentaries. This may be dismissed as nonsensical: why would a blind person want to watch documentaries. Yet in many ways of course it does make sense: A video documentary is not only visual, there's a lot of auditory information in documentaries as well. And of course, apart from the documentary itself there's

a lot of extra information about the documentary as well, like the description and meta information that's of interest to Simon's research as a designer.

It reminds me of a discussion I had with Katrien Vermeulen, my wife, a long time ago when I still worked at this web design agency. Back then we were working on checkout flows for webshops, and our sole focus was on making it as easy and as frictionless as possible for people to buy stuff. Making it harder to buy things sounded like utter nonsense to me. Until I discussed this with Katrien who works with homeless people and people with severe, chronic debt issues. She convincingly argued that there are very good reasons to make it harder instead of easier to buy things.

Both these examples, the one of making videos accessible to blind people, and the one about making webshop checkout flows harder to use are good examples of not dismissing arguments that sound nonsensical without giving them serious thought. Allowing this kind of nonsense into our research can broaden our vision and can result in completely different products.

Invisible animations

And then there are the completely unexpected ideas that may pop-up when you ask for nonsense. Like the project with the invisible animations I did with Hannes Wallrafen.[41] When I asked Hannes if he had some ideas that were too ridiculous to suggest, he told me he wanted funny little animations on his website, even though he is blind. Animated illustrations aren't very weird, and even coming up with a textual alternative is something we've done before. But by designing the animations first for Hannes, and testing them first with him I came to new insights. In hindsight it's almost too obvious, but it's a very clever prototyping workflow to first test the textual alternative of an animation before you create the animation itself. Creating an animation is a lot of work, writing a small punchline is much easier to create, and much easier to iterate.

Sniggering screen readers

Another idea that came out of this project was the idea to make screen readers laugh. It was the first time Hannes heard some emotion from his screen reader. It made me realise that even within the tight constraints of a screen reader, which sounds neutral and maybe even boring by design, we are able to add some emotion to our design.

This idea of making screen readers laugh resulted in the open database of strings that make different voices of different screen readers laugh. This database is funny, sure, but it is also a first step in trying to lift screen reader interaction

beyond what Graham Pulling calls engineering, into the field of design. You are very welcome to contribute.

Fun

Of course there are serious reasons for allowing nonsense into your design process. But for me the most important reason for using nonsense, and not some other ideation method, is to allow *fun* into the design process. Designing accessibility, helping disabled people lead an independent life is serious work. But it can be fun as well. And only if we allow fun will we be able to start making pleasurable user interfaces. If we do, maybe some day in the future Léonie Watson will come over to my university again, and this time she will be able to explain what makes an interface fun to use for someone who is blind.

And so, by writing about nonsense, we've reached the end of this thesis. The only thing that’s left is a conclusion.

Conclusion

You have reached the last chapter of my Master thesis, dear reader. All that's left for me to do is present you a with a clear and ambitious conclusion.

What did this design research teach me?

By using my Exclusive Design Principles to design tailor made solutions for real people with real disabilities I flipped the ability bias. And as my own ability and knowledge was no longer the starting point everything changed. Assumptions turned out to be untrue, best practices turned into dark patterns, and nonsensical ideas turned out to be useful. But most of all it did prove one of my assumptions true: UX for people with disabilities is indeed very poor.

In my research I have only worked with a handful of people with a few disabilities. There are so many more people who have been excluded systematically. Just imagine what it is like to use podcasts when you're deaf, to get information from your municipality when you have a lower IQ, or when you have to browse the web when you're motor disabled. To name just a few situations in the context of my niche. Right now these situations are mostly treated as insignificant and exceptional edge cases. Yet a completely new field of design opens up if we realise that we can really improve the lives of so many people if we start involving them in our design. The field of accessible design is vast, and it is wide open. There is so much research to be done in this niche of specialist user interfaces for individual needs.

Study situation

Working with the first of my exclusive design principles — Study situation — showed me that in many cases the situation is even worse than I already expected. The web is not merely *hard* to use for people with disabilities, in many cases it is *impossible* to use. Even if a website is somewhat functional, the user experience for someone who uses assistive technology like a screenreader is unimaginably poor. Comparing it to the visual web as most of you probably know it, is like comparing crossing the ocean in a rubber boat to flying over it in an airplane. I don't believe such an enormous difference in experience is necessary.

In part the huge inequality is due to how screen readers work. The learning

curve for using these tools is very steep, and basic knowledge is not enough for regular users. It seems like these tools are mainly designed for expert users. And even when experts use them, unexpected things happen all the time. At the moment they lack any of the refinement and easy of use that we have in our visual environments. A lot of research can be done into how to improve these tools to make them more accessible and less frustrating to all people who depend on them.

Unsurprisingly, the web is almost exclusively built by people who don't know anything about accessibility. This is to be expected for smaller websites, made by amateurs, but surprisingly the same problem exists in very large organisations as well. I've observed blind people trying to order groceries at the largest super market in the Netherlands, and trying to transfer money at the largest bank. All to no avail. A clear hierarchical overview is often lacking, and very often core tasks cannot be found because there's simply too much stuff on webpages. I would expect serious design teams to adhere to much higher standards when it comes to inclusion. What these higher standards look like is unknown. There is much research needed into what inclusive design means for large organisations and for professional design teams.

Ignore conventions

The second principle of exclusive design — Ignore conventions — is an important one. Many web designers and developers assume that most UX patterns have been invented by now, and that copying and pasting things from other websites will result in a good enough experience. While it is easily debatable if this would be true for the visible web, my design research has shown me that it is clearly not true for people with disabilities. I have found that many of the patterns that we take for granted — like the navigation at the top of every single web page — make no sense at all from a blind person's perspective.

I have found ignoring conventions to be an easy and rewarding exercise when it comes to designing for people with disabilities. Replacing confusing patterns with something else, or simply removing stuff that we have come to think is necessary quickly improves the user experience.

Again, I have only touched the tip of the iceberg in the little research I've done so far, there is so much design research to be done in this field of accessible user interfaces.

Prioritise identity

I have taken the first exclusive design principle, of studying situations, further.

Instead of simply observing, I have actively worked together with individual people with disabilities. The tailor made solutions I made were not created by the sublime genius of a *Grande Artiste*, on the contrary. They were a logical collaboration between someone who knows what it's like to be excluded based on ability, and a designer who has detailed knowledge of the possibilities of the web.

This collaboration is important. By simply observing Simon Dogger, a blind product designer, I wouldn't have decided to do a tailor made redesign of a website with TV documentaries. And I would never have come up with the idea to create *invisible animations* without working together with Hannes Wallrafen, a blind artist.

I have worked with this principle on a small scale, creating exclusive solutions for individual people. This helped two or three people. The impact of involving excluded people in the design process would of course benefit many more people.

Both the idea of opening up the industry to designers with disabilities and the idea of involving people with disabilities into the design process are relatively new. Indeed, also in this field there is so much room for research.

Add nonsense

A tool that turned out to be very helpful in forcing innovation was using the principle of *Adding nonsense*. Allowing all kinds of ideas into the brainstorm process is important, especially when designing for such a relatively unknown field as inclusive web design. Ideas that may seem ridiculous could very well turn out to be valuable. This turned out to be true in my research.

It started with the nonsensical idea of working with the exact opposite of a set of very sensible principles. This silly idea turned out to be the basis of this whole master research. And forcing my students, my co-designers and myself to add nonsense resulted in some of the better ideas. Without nonsense there wouldn't have been a six key typing tool, and I wouldn't have tried to make screen readers laugh.

My ambitions

Many designers move away from the web. They think it's boring. And I have to agree with them. If your job is copying design patterns made by others into a yet another mediocre project, that sounds like an unrewarding job to me indeed. There are emerging technologies that look much more interesting. I would like to propose that we call inclusive design an emerging technology as well. It's

new, it’s largely unknown, it’s fun, there’s a lot to innovate, and it’s very rewarding work. And it’s really necessary.

My ambitions when I started this research were rather low. I didn’t expect inclusive design to be such a complicated field. My first idea was to create modules for inclusive design education. The idea was that these modules could be plugged in to all kinds of design classes. The idea of educational modules is still a good one, and someone should definitely work on it. But quite early in my research my ambitions changed.

All accessibility experts will tell you that one of the reasons why the level of inclusive design is so low is because inclusive design is not taught at design schools. In that light my initial idea is not such a bad one. But when I look at all the open plains where no design research has been done, I can only conclude that there is room for better inclusive design education. Of course I, and some of my colleagues will continue to work with inclusive design at the CMD digital interactive design school on a bachelor level. This will create at least a few future designers for whom inclusive design is not an edge case, but a logical first step.

These bachelor students will only in part be able to do something about the growing and urgent need for expert inclusive designers. Inclusive design expertise is needed now that all new government websites *must* be accessible. But practical expertise alone is not enough to lead such an enormous operation. Inclusive design leadership is needed as well.

This means that there should be *a new Inclusive Design Master* here in the Netherlands. Such a master could provide room for the much needed research that needs to be done. And it could deliver the leaders the industry needs to make the transition from exclusive to inclusive design.

Such a master could play an important role in the last wide open plain that needs thorough exploring: the fact that our schools are not accessible. At least, the one where I work. We have one blind student, and to our horror most of our classes and assignments are impossible for her to do. Our slides are inaccessible. Our assignments are strictly visual. Our tools are impossible to use. Similarly, Marijn Meijles dropped out of university because he had to write papers. Marijn was graded for his typing skills, which are low since he’s severely motor disabled.

Now that I am done with my Master Design research I am going back to what I like most: Teaching at the CMD in Amsterdam, but this time with an increased focus on inclusive design. And next to teaching I will do my very best to make this new Master Inclusive Design happen.

— Vasilis van Gemert, 2019.

Colophon

- This website is made by **Vasilis van Gemert**
- **Exclusive Design** is the title of this master thesis
- This thesis was published in January 2019
- The research was done at the Design Master at the Willem de Kooning Academy in Rotterdam
- I’d like to thank **Harma Staal** for her invaluable coaching
- Fonts I’ve used are Greta Sans and Greta Display by Peter Biľak.
- The text was typed using Markdown.
- The website is tailor made, by hand, using HTML, CSS, JavaScript, PHP, and fun.

Literature

I have been inspired by an endless list of articles and presentations that I have read and seen throughout my career as a web professional. I have also been inspired by a list of books that should be longer. So alas, on this page I cannot mention all the literature that has insipired me, there is simply too much of it. Yet here is the list of books, articles and presentations that had a direct influence on this design research.

Books

- Graham Pullin. Design Meets Disability. MIT Press. 2009
- Kat Holmes. Mismatch, How Inclusion Shapes Design. The MIT Press, Hardcover. 2018
- Sara Wachter-Boettcher and Eric Meyer, Design for Real Life, A Book Apart Publishers, New York, ePub Edition, 2016
- Hannes Wallrafen. De Blinde Fotograaf. Uitgeverij Atlas Contact. Paperback. 2018

Websites and online articles

- Henny Swan, et al. “Inclusive Design Principles.” Inclusive Design Principles, Pacielo Group, 2017, inclusivedesignprinciples.org/
- W3C Mission, Design Principles. Website. Accessed on 16 December, 2018. www.w3.org/Consortium/mission.html#principles
- Robert Jan Verkade. Ik maak het uit met de Webrichtlijnen. Archive of website. 2012. web.archive.org/web/20121028045622/idontcareabout.net/

Presentations

- Léonie Watson. What does User Experience mean when you’re blind? Icons Meetup. 2017. vimeo.com/222490579
- Brad Frost. Death to Bullshit. Creative Mornings Meetup. Presentation. 2013. vimeo.com/63437853
- Johan Huijkman. UX candy. Fronteers meetup. Presentation. 2013.

fronteers.nl/bijeenkomsten/2013/q42
- Marie van Driessche. Ontwerpen voor Doven, Design by Fire Café. Presentation. 2017. designbyfire.nl/cafe/062
- Stephen Hay. I don’t care what Airbnb is doing. (And neither should you. Conference talk. 2018. noti.st/stephenhay/nLABeP/

Thank you

Thank you to Hannes Wallrafen, Larissa Klaassen, Marie van Driessche, Marijn Meijles, and Simon Dogger for making this research real.

Thank you to the students of the 2018 minor web development, and especially to Bas Kager, Jamal van Rooijen, James Peter Perrone Jefferies, Jelle Overbeek and Rick Buter, for designing real exclusive interfaces for real people.

Thank you Léonie Watson for coming over to Amsterdam to give a lecture at one of the Icons meetups at the University of Applied Sciences. Without realising it you kicked off this whole research back then.

Thank you to all of you who took part in one of the Exclusive Design Challenges.

Thank you to Alla Kholmatova, Bram Duvigneau, Fransiska Groenland, Heydon Pickering, Irene Kamp, Ischa Gast, Johan Huijkman, Justus Sturkenboom, Koop Reynders, Laura Kalbag, Léonie Watson, Maarten Kappert, Marrije Schaake, Michèle van den Aardweg, Paul van Buuren, Peet Sneekes, Peter van Grieken, Reinier Ladan, Robert-Jan Verkade, Ron Kersic, Sanne ’t Hooft, Stephen Hay, Timon van Hasselt, Titus Wormer, Vitaly Friedman, for making this a better thesis, and making me smarter.

Thank you to Joschi Kuphal, Kevin White, Marc Thiele, Peter Heery and Stefan Judis for inviting me to talk about my research at your conferences.

Thank you Deanna Herst, Hanneke Brier, Harma Staal, Jan Belon, Jop Japenga, Sietske Klooster, and all the guest lecturers for your inspiring classes.

Thank you to my wonderful fellow students, with their own wonderful niches. I am going to miss our bi-weekly Chinese food, our conversations and our shared frustrations. Barend Onneweer, Joost de Boo, Jurgen Wiegeraad, Manon Mostert-van der Sar, Mariska van Zutven, Michelle Baggerman, Wiesje Korf, Yuri Westplat — and all the dropouts! — you have all inspired me immensely!

Thank you Yuri Westplat for our *Statler-and-Waldorf-esque* conversations about design on the rides back to Amsterdam on Thursday evenings.

Thank you to all my guests at my The Good, The Bad, and The Interesting podcast. I thought my master research would be about our conversations, which is still a good idea.

Thank you to my colleagues at CMD Amsterdam for creating the best working environment ever.

Thank you to Harry Zengerink who blackmailed me into doing this master.
Thank you to Diek Kubbe and Joke Werkhoven, my teachers, for waiting with moving to the countryside until after I finished my master.
Thank you my dearest Katrien Vermeulen and Kiki van Gemert, I love you.
And thank you for reading.

Tools, and websites

Here is a list of all the tools, examples and websites I have created during my Exclusive Design research:

- The semantic responsive illustration of circles.
- Visualise semantic clutter with the Screenreaderizer
- An example of focusing with arrow keys
- Typing with six keys prototype (source code)
- 2 Doc for Simon, application. Download for Linux (zip 52MB, untested), Mac (zip 50MB), of Windows (zip 53MB, untested)
- The sound projects of Hannes Wallrafen
- Cheer reader, documenting strings that make screen readers giggle.

---

1. More on these principles in the chapter called Flipping things ↩

2. I have grouped the articles that dig into this question on this page called Study Situation ↩

3. The articles in which I explore this question are grouped under the moniker Ignore conventions ↩

4. The articles that document this idea of actively involving people with disabilities in the design process are collected in the theme Prioritise identity ↩

5. The articles about this question are grouped in the fun sounding principle Add nonsense ↩

6. W3C Mission, Design Principles. Website. Accessed on 16 December, 2018. www.w3.org/Consortium/mission.html#principles ↩

7. Peter-Paul Koch et al. Guild, part 1 - certification. Blog post and comments. 2007. www.quirksmode.org/blog/archives/2007/07/guild_of_fronte.html ↩

8. Mike Monteiro. Design's Lost Generation. 2018. Blog Post. medium.com/@monteiro/designs-lost-generation-ac7289549017 ↩

9. See my public conversation with Bram Duvigneau ↩

10. See the chapter called More Death to More Bullshit ↩

11. See the chapter Semantics Schmemantics ↩

12. Kat Holmes. Mismatch; How Inclusion Shapes Design. Hardcover. The MIT Press. 2018 ↩

13. Swan, Henny, et al. "Inclusive Design Principles." Inclusive Design Principles, Pacielo Group, 2017, inclusivedesignprinciples.org/ ↩

14. Kholmatova, Alla. "From Purpose to Patterns." CSS Day. CSS Day 2018, 14 June 2018, Amsterdam. ↩

15. See my public conversation with Bram Duvigneau ↩

16. Winners of the Ig® Nobel Prize. Improbable Research. Website. 2018. improbable.com/ig/winners/ ↩

17. Eva Westerhoff. Tweet. 2016. twitter.com/evawesterhoff/status/774107689622175744 ↩

18. Marie van Driessche. Ontwerpen voor Doven, Design by Fire Café. Presentation. 2017. designbyfire.nl/cafe/062 ↩

19. Where's that example about a blind father buying a car for his kid? ↩

20. Bram Duvigneau in gesprek met Vasilis van Gemert. Online video. 2018 youtu.be/7o-O7L6UJ7c ↩

21. Brad Frost. Death to Bullshit. Creative Mornings Meetup. Presentation. 2013. vimeo.com/63437853 ↩

22. Vasilis van Gemert. The Screenreaderizer. Bookmarklet. Github Page. Visited on January 3, 2019. vasilisvg.github.io/screenreaderizer/ ↩

23. When I counted on the 4th of January 2019, there were *at least* 418 HTML

elements that have some semantic meaning which gets spoken out loud by Simon's screen reader. ↩

24. Wallrafen, Hannes. De Blinde Fotograaf. 2018. Uitgeverij Atlas Contact. Paperback. ↩

25. Campbell, Alistair et al. Web Content Accessibility Guidelines (WCAG) 2.1, W3C Recommendation 05 June 2018. website. www.w3.org/TR/WCAG21/ ↩

26. Holmes, Kat. Mismatch, How Inclusion Shapes Design. The MIT Press, hardcover edition, 2018. Page 61. ↩

27. You can see for yourself if the tailor made website I created for Hannes Wallrafen works for you. ↩

28. Léonie Watson. Understanding semantics. Blog post. 2016. tink.uk/understanding-semantics/ ↩

29. Anne van Kesteren, Maciej Stachowiak. HTML Design Principles. 2009. dev.w3.org/html5/html-design-principles/ ↩

30. Sara Wachter-Boettcher and Eric Meyer, Design for Real Life, A Book Apart Publishers, New York, ePub Edition, 2016 ↩

31. Stephen Hay. I don't care what Airbnb is doing. (And neither should you. Conference talk. 2018. noti.st/stephenhay/nLABeP/ ↩

32. Graham Pullin. Design Meets Disability. MIT Press. 2009 ↩

33. Holmes, Kat. Mismatch, How Inclusion Shapes Design. The MIT Press, Hardcover ↩

34. Six Key Typing. Rick Buter and Vasilis van Gemert. Git repository. Accessed December 16 2018. github.com/vasilisvg/six-key-typing ↩

35. Creative Quest. Questlove. Ecco. Hardcover. 2018 ↩

36. Human Colours. Vasilis van Gemert at al. Git Repository. 2013. github.com/vasilisvg/human-colours ↩

37. See for example Seeing AI , an iPhone app by Microsoft. ↩

38. Maaike van Cruchten et al. The First Exclusive Design Challenge. Team Larissa. 2017. vasilis.nl/gbi/exclusive-design-challenge/#teamlarissa ↩

39. See the chapter about false assumptions called Fuckup's mama ↩

40. See the chapter about Designing Like It's 1999 ↩

41. See the chapter about Insivible Animations ↩