**Immagine che contiene testo, Carattere, Elementi grafici, logo

Descrizione generata automaticamente**

**WEB INFORMATION MANAGEMENT**

**2023/2024**

**Prof. *Massimo Marchiori***

Written by *Michael Amista’*

**Summary**

[**1. Introduction to a website** 3](#_Toc148623524)

[**1.1 The problem of “time”** 3](#_Toc148623525)

[**1.2 The importance of structure** 7](#_Toc148623526)

[**2. Usability problems** 8](#_Toc148623527)

[**2.1 Persistent problems** 8](#_Toc148623528)

[**2.2 Non persistent problems** 10](#_Toc148623529)

# **1. Introduction to a website**

## **1.1 The problem of “time”**

There’s a parallelism between a website and a shop. In both there’s a window (homepage) that shows what the website/shop contains/sells.

What do the people expect from a homepage? Many things, but the most important is the **information**. The problem of information synthesis has not invented with web pages … it’s a problem common in *information communication*, the problem was born with journal articles.

The most important information components of a website homepage: **informative axes**:

* **WHERE?** Where did I (user) arrive?
* **WHO?** Who is behind the website?
* **WHY?** What are the benefits? Why should I stay?
* **WHAT?** What choices do I have?
* **WHEN?** What are the last news?
* **HOW?** How do I arrive to where I want?

That’s the **core of informatio**n, what the people want to know. If these parts are missing the information is uncompleted. Users want to know this information.

We need to convince the user to surf on our website.

**But** that’s not so easy to fill the homepage with this information. There’s a “small” problem… **TIME!** Users have expectations and have **limited time**. It’s easy to put everything on a webpage but people don’t have the time to read/see everything.

A user arrives to our homepage and he spend, on average, **31 seconds**. So, we have on average 31 seconds of time to convince the user to stay in our site and show him the information components. And the way to do it is to **compress the information**.

For example, *how much text can we put in our home page?* It depends on how much faster the people read but on average from **200 to 300 words per minute which go down to 180 in case of computer screen reader** because it reads slowly than a human.

So, if we put more than 93 words we have finished our time. They should be quite less, because the user doesn’t spend all the time just to read, but there is also time lost to analyse the visual layout, images, links and more.

**We want not only he coming to our site, but we also want him to come back!** So, what about the returning user? Same expectations? He knows you already, so the limited time for the home page will not used the axes WHERE, WHO and WHY. The time will be spent only on the remaining axes, WHAT, WHEN and HOW.

**The disadvantage of returning is more demanding**, and he has less time to spend for us.

Time on homepage: first visit (31 seconds) / second visit (25 seconds) / third visit (22 seconds) …

So, if we want to make our returning users happy, we have a little treasure of **19 seconds** four our homepage, to split among the WHAT, WHEN and HOW parts. That is a maximum of **57 words for all these components**. He will skip the parts that he already knows.

I must open a website and obtain the main information from the webpage, try with this:



**Too hard! It’s not clear!** We can do better:

Immagine che contiene testo, schermata, Sito Web, Pubblicità online

Descrizione generata automaticamente

**And after the home page?** If we have success the user goes on to the other pages of the site so we cannot put everything into the homepage.

**What must we care about in the other pages?** Same thing of the homepage? NO, the user already knows where he is, so no need anymore for all the WHERE, WHO, WHY, WHAT, WHEN, HOW axes.

Once we attracted someone inside our site, we have the advantage that typically they are less likely to go away. On average, **the returning user stays more time**. From the 31 seconds of the homepage, we pass to **53 seconds**! That’s because the user entered in the shop/site, he looks at the window/homepage and now he’s inside.

This additional time allows us to add information, so having pages that are more specific than the homepage. But always remember to pay attention, and don’t exaggerate: there are always the max limits on text 🡺 53 seconds corresponding to **159 words**.

For instance, that’s the old version of the Apple website, the red box contains the 159 words (53 seconds) that the user read and then the time is over! User doesn’t know the advantages of the new iPod because that information is much later in the text.

Immagine che contiene testo, schermata, Sito Web, Pagina Web

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Sito Web, Pagina Web

Descrizione generata automaticamente

The site has been redesigned, a lot of information to share? ***Learn more…*** after every section. It’s a user choice to know more about a feature, you’ll redirect to another webpage with another 53 seconds to spend. Every page must be designed on 53 seconds of reading. **The user choose what to do!**

There is another time that is very important for the users. **The global time is the time it takes the user to be satisfied: he found what he wanted on our site**. For instance, the global goal is to buy a new iPod and the global time is the time I spend to buy the iPod. The global time goes over the single page, it’s the experience.

The first global time is the time by which the user has got an idea on us and decides whether to stay with us (to the final goal) or to go. It’s the so-called **choice time**. The choice time is **1 minute and 49 seconds**.

When a user goes away because he’s not satisfied, we probably (88%) lose him forever. Have the best price or product is not enough, the user will go away anyway if he’s not satisfied.

The second global time is when the user expects to have found what he wanted, and so to have successfully navigated our website. This limit is **success time** and it’s about **3 minutes and 49 seconds**.

**So, what matters most is the good balancing among *homepage, internal pages* and *trail (path)* that the user follows to arrive where he wants**.

Moreover, given the success time (3m 49s), he expects to reach his goal after having seen the homepage and about three pages and a half of our site.

## **1.2 The importance of structure**

For every goal we can build the best path for the user to achieve it. The **tree structure** of our site then becomes critical, as well as the distance from the home page: from the homepage, after **one click** (max 2) we have to convince the user, and after other **two clicks** (max 3) we must give them what they wanted! We have to offer shortcuts to the goal.

The structure of the web has changed: whereas time ago navigation always usually started from the homepage, nowadays this not true anymore 🡪 to search x, navigation can start from any place (Google), this is called **deep linking**. The user can see an internal page before the homepage and he cannot see the homepage never. For us this is a challenge!

The situation is therefore more complex: **each page can be the first page that a user sees**. So let’s see what happens to the axes: some axes become mandatory, others optional, and others can be omitted.

**Mandatory informative axes**:

* **WHO** (typical shortcut: logo in the upper-left part)
* **WHAT** (typical shortcut: direct link to the home page)

Optional informative axes:

* **WHEN** (completely optional)
* **WHY** (suggested!) == short description
* **HOW** (suggested!) == typical shortcut: the search functionality (preferred position up-right part). Suggest to insert correlated pages.

The most important informative axes is **WHERE** because the user is “thrown” in the middle of our information forest 🡪 typically we should make clear the context.

Why WHAT is not enough? To avoid for the user to always go to the homepage, wasting a click. **If the user directly lands on a certain page, we have more information on what he wants!** Let’s use that information, instead of pushing him back to the home page.

*Example*: if a user entered in a shopping page where he can buy a tv we can put a shortcut like “Do you want see other electronic products?” to avoid the user goes to the homepage.

**Immagine che contiene testo, schermata, tenda

Descrizione generata automaticamente**

**Typical error: put the homepage menu in a specific page**.

The user doesn’t care about the global functionalities in a specific page. He prefers knows what he can do inside that page and for the entire menu I can put a shortcut to the homepage.

These techniques, that explain to the user where he is, are called **breadcrumb**. Three main kinds of breadcrumbs (we can use just one or all three):

* **Location**: where we are (Home >> News >> News x). It helps the people to navigate in the website;
* **Attribute**: shows the attributes of the given page, so it does not necessarily correspond to the location, but contains the path from the main type and its subtypes (for example a component sales site may have HW / graphics cards / nvidia / geforce). Each page has tags that help understand the categorization;
* **Path**: show the path taken by the user to arrive to the page. Typically they are dynamical.

The classic separators for breadcrumbs are two: “>” and “/”.

# **2. Usability problems**

Can be divided into two main classes: **persistent** and **non-persistent**. The persistent problems are those that didn’t change much along time (typically, the worst problems). The non-persistent ones, instead, did change (typically, for the better).

## **2.1 Persistent problems**

When navigating inside a site, an important problem that needs special care is the **lost in navigation**. Users must be always conscious of where they are within the site, so WHERE axis.

We also see the breadcrumbs but we can add something. Navigation is not only where I am but als where I’ll go. **To know where to move next, we also need to know where we have been**. Sure we can just suppose a user to remember where he has already been. But doing so, we pass the burden to the user: remembering this information **makes navigation heavier 🡪** we have to satisfy the user.

For instance, we can use a functionality that is present in browsers since a long time: **change color to links that have been already visited**. Giving more powers to designers, many just forgot why this functionality was introduced 🡪 designers don’t have idea of the rules of usability, they just use the more elegant thing. So a big usability problem arose: **don’t change colors to already visited links!**

**Paradox**: user that like our site will navigate more but the more they navigate, the more we ask for them to remember the pages they visited. So, the more they navigate, the more they must memorize, and to get tired. *[It’s like a customer must do the work of a casher where he goes to the supermarket 🡪 he must remember everything he put in the basket and after he has to pass the items to the counter 🡪 it’s a stressful experience!]*

So, users must be able to move quickly, ***what are the navigation movements that are most used?***

1. **clicking on a link** to go to another page;

2. pressing **back!** Users prefer to navigate back even many times, **even when there is a direct link**. They use it even up to 7 times (7 clicks), instead of doing one direct click 🡪 designer can forget about this functionality.

How so? Instead, users are wasting time? Yes, it’s paradoxical, but there is a reason. Users are still **minimizing** the computational effort. The advantages of the back button are related to no need to remember the followed path and the interface is consistent for every site.

So, the corresponding usability problem is then **not to allow the proper use of the back button**. Usually happens with dynamic pages that are badly handled, not saving the navigation state in a way that is compatible with the back button.

Another possible interference, and related severe usability problem, is to **open a new browser window**. Many designers use new windows to try clearly separate new content. Opening a new browser window gives a few problems to the user.

The first bad problem is to **disallow the use of the back button**, with all the consequences seen above the navigation.

New windows can typically be of two kinds: a new browser window, or a new tab. They both irritate and confuse users, although the worst one is the first (new window).

The worst problem, in this case, is that the page stays above the existing navigation 🡪 a lot of windows opened that can hidden the other ones opened.

A related problem is the **pop-up**: they are opened **without the user consent**. We will come back later but this type of window is so hated by users to deserve a special mention as a standard alone usability problem.

Another big usability problem is to **violate web conventions**. A web convention is not a web standard but just a technique used by most web sites. **Respecting conventions is related to one of the most famous usability laws**. New methods, never used by nobody, can be better but it takes time and effort to be understand from the users.

Users spend most of their time on other websites so we should not bend users to our will if they are used to other things on the web. If we do otherwise they need to adapt to our different design 🡪 that takes time, energy and leads to frustration and timers expiration.

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamenteUsing too much **empty language or languages with little content and lot of slogans** 🡪 that’s no propaganda (what woks on TV not always works on websites).

[*home page of montblanc pens*]

Another related problem is the *form*. One of the more severe usability problems is to **use difficult and monholitic text**. This often happens also in sites that can ignore user timers and their feelings, for istance public/governmental sites, or private but monopolist.

In general web text is different than normal text: besides timers, reading on a screen is more difficult 🡪 text should be simplified, so to counterbalance the additional effort.

**The rules of web text**:

- Base rule: 100% normal text 🡪 50% web text

- If generalistic audience: 100% normal text 🡪 25% web text

- It’s helpful to start with the conclusion (synthesis of information: I have knowledge of what we are talking about and then I can decide to go on or no) and then expand

## **2.2 Non persistent problems**

The *non persistent* problems are those that changed in time.

One of this kind of problems are the **splah pages**.

Immagine che contiene testo, Mongolfiera, schermata, logo

Descrizione generata automaticamente

A type of intermedium page between us and the website’s home page.

**Avoid at all cost**: besides not being liked by users, t**hey make them lose precious time**.

Consequence: even worst if animated 🡪 it’s non persistent because the trend is changing with faster internet people can think to put everything on this page and this wastes user’s time.

Another problem is **asking personal information**.

Immagine che contiene testo, software, Software multimediale, multimediale

Descrizione generata automaticamenteInstance of the problem is *premature registration*.

**We haven’t already seen the website and they ask us our information!?**

Another examples are the ***pop-up*** that shows, for example, when we are not already registered in the site.

These are to avoid: **how much time is the user losing to give information? The time doesn’t stop when a pop-up appears, goes on!**

In the case of premature registration, there is also the additional disadvantage of the registration, which implies a further **computational effort** to insert and then remember a data pair made by login and password for that service.

Last but not least, the **trust problem**: giving away personal information requires a *trustworthy* site, and in case of premature registration the trust bond has still to be established!

🡺 premature registration brings a decrease of potential users (on average less than 1 out of 10).

Immagine che contiene testo, diagramma, linea, Diagramma

Descrizione generata automaticamenteAnother problem is **scrolling**. On average, how much users are willing to scroll? 1.3 screens (so seeing a total of 2.3 screens) 🡪 they hardly see what is beyond, and when they do their effort increases.

**Mostly of the people just see the content of the first scroll**.

*N.B.* the mobile rules are different because the screen is “longer”.

**Are there difference between home vs internal pages?** Yes:

* First visit to the home page: 23% scrolls
* Internal pages: 42% scrolls
* Repeated visits to the home page: 14% scrolls

Immagine che contiene software, schermata, testo, Software multimediale

Descrizione generata automaticamenteThe instance of iPod site is perfect: the standard size of a web window is **1024x768** and what happens is the image takes a lot of space 🡪 that could be taken by more important information.

**Bad attitude**: put oversize image that takes a lot of space.

Immagine che contiene testo, schermata, multimediale, software

Descrizione generata automaticamenteUsers are forced to scroll to see what there is beyond these oversize images.

From laptop to net-books’ reference size: 1024x600. But what is the max size? The average *safety size* to consider is 800x600.

The problem of scrolling is**: too much information expanded on the vertical screen axis**. What about the **horizontal**? When information clashes with the horizontal axis, we have the **frozen layout problem**.

We scroll vertically, so often the horizontal axis is just “frozen”, giving a set width or a (too large) minimum width 🡪 we have to move the horizontal axis to see beyond the frizzed part. This way, **then the window size goes beyond these parameters we have serious visualization problems**.

For instance, in the old iPod site we have the frozen layout problem and the hidden components are fundamental, like the search bar which is so important for a user.

When the window is smaller than the set/min size (not full screen) then things go wrong for lack of space. In this case to access horizontal information the users need a **horizontal scrolling**. That is one of the most hated things by users 🡪 **it is uncommon** on the web, and it is not within our classic information modalities.

In our life, we have always been used to handle textual information with one leading dimension: books, screen pages etc 🡪 in these systems, the horizontal axis is always set.

This problem also augments the **dimensionality** of the information space: 2 dimensions to manage! 🡺 more computational effort!