

# Appunti di Tecnologie Web 2 - v1.00

Eduard Bicego

14-04-2016

*“Agli abitanti dello SPAZIO IN GENERALE  
Questo lavoro è dedicato  
Da un umile nativo di Terrapiatta  
Nella speranza che  
Anche se egli p stato iniziato ai ministeri  
Delle TRE dimensioni  
Avendo finora convissuto  
Solo CON DUE  
Così i cittadini di quella regione celestiale  
Possano aspirare ancora e sempre più in alto  
Ai segreti delle QUATTRO CINQUE O ANCHE SEI dimensioni  
In questo modo contribuendo  
Allo sviluppo dell’IMMAGINAZIONE  
Tra le razze superiori  
dell’UMANITÀ SOLIDA”*

*Edwin Abbott - Flatlandia*

## Abstract

“ ”  
.

# Contents

<b>1</b>	<b>Primi passi nel Web (*)</b>	<b>8</b>
1.1	1945 Memex . . . . .	8
1.2	1960-68 NLS: onLine System . . . . .	8
1.3	1960 Xanadus . . . . .	8
1.4	1980 Enquire . . . . .	8
1.5	1990 World Wide Web . . . . .	8
1.6	Le alternative al WWW . . . . .	8
<b>2</b>	<b>Il sito Web</b>	<b>9</b>
2.1	La metafora del negozio . . . . .	9
2.2	Problemi e implicazioni . . . . .	9
2.2.1	Quanto testo nella homepage? . . . . .	10
2.2.2	Il comportamento dell'utente è dinamico . . . . .	10
2.2.3	Questione di tempo . . . . .	10
2.2.4	E il resto del sito? . . . . .	10
2.2.5	Questione di tempo - Parte II . . . . .	11
2.3	L'importanza della struttura . . . . .	11
2.3.1	Deep linking . . . . .	12
2.3.2	Gli assi in dettaglio . . . . .	12
2.3.3	L'importanza del Where . . . . .	12
<b>3</b>	<b>Problemi di usabilità</b>	<b>14</b>
3.1	Problemi persistenti . . . . .	14
3.1.1	Navigazione . . . . .	14
3.1.1.1	Dove sono, dove ero e dove sarò . . . . .	14
3.1.2	I movimenti dell'utente . . . . .	15
3.1.3	Nuova finestra? No, grazie . . . . .	15
3.1.3.1	Un problema correlato: i pop-up . . . . .	16
3.1.4	Convenzioni violate . . . . .	16
3.1.5	Altri problemi: What non rispettato . . . . .	16
3.2	Problemi non-persistenti / Il contenuto . . . . .	17
3.2.1	Splash page . . . . .	17
3.2.2	Richieste di registrazione . . . . .	17

3.2.3	Lo scrolling maledetto . . . . .	17
3.2.3.1	Taglia dello schermo . . . . .	18
3.2.3.2	Scrolling orizzontale . . . . .	18
3.2.3.3	People do scroll . . . . .	18
3.2.4	Lo sforzo computazionale spiegato da Engelbart . . . .	18
3.2.5	Bloated design . . . . .	19
3.2.6	Abusi multimediali . . . . .	19
3.2.6.1	Il 3D - Prima, dopo, ora . . . . .	19
3.2.6.2	I plugin . . . . .	20
3.2.6.3	Dai plugin a Flash! . . . . .	20
3.2.6.4	I video . . . . .	20
3.2.7	La Metafora visiva . . . . .	20
3.2.8	I menu di navigazione . . . . .	21
3.2.8.1	Pathfinding . . . . .	21
3.2.9	Il testo . . . . .	22
3.2.9.1	Caps lock . . . . .	22
3.2.9.2	Immagini sostitutive . . . . .	22
3.2.9.3	La maledizione Lorem Ipsum . . . . .	23
3.2.9.4	Scanning . . . . .	23
3.2.9.5	Strutturazione . . . . .	23
3.2.9.6	Blonde effect . . . . .	24
<b>4</b>	<b>Usabilità: Siti E-Commerce</b>	<b>25</b>
4.1	Il prezzo . . . . .	25
4.1.1	Trappola iper-associazione . . . . .	25
4.2	La pubblicità . . . . .	26
4.2.1	Pubblicità classica e tecniche . . . . .	26
4.2.2	Noi e la pubblicità . . . . .	26
4.2.3	La pubblicità nel web . . . . .	26
4.3	Il prodotto . . . . .	27
4.3.1	Descrizione del prodotto . . . . .	27
4.3.2	L'aspetto visivo del prodotto . . . . .	27
<b>5</b>	<b>Il comportamento degli utenti</b>	<b>28</b>
5.1	Un media classico: Il giornale . . . . .	28
5.1.1	L'importanza delle immagini . . . . .	28
5.2	Il web . . . . .	28
5.2.1	Fasce d'attenzione . . . . .	29
5.2.2	L'importanza del testo . . . . .	30
5.2.2.1	Paragrafi e titoli . . . . .	30
5.2.3	Pagine grasse o magre? . . . . .	30
5.2.4	Immagini nel web . . . . .	31
5.2.4.1	Slideshow . . . . .	31
5.2.5	Lo spostamento dell'utente: Fitts . . . . .	31

5.2.5.1	Legge di Fitts . . . . .	31
5.2.5.2	Interfacce e Fitts . . . . .	32
<b>6</b>	<b>La pubblicità</b>	<b>35</b>
6.1	Posizionamento . . . . .	35
6.2	Messaggio efficace . . . . .	35
6.2.1	Top 11 disgrazie . . . . .	36
6.2.2	Il mondo pubblicitario nel web . . . . .	36
6.2.2.1	Effetto zapping . . . . .	36
6.2.3	Immagini in serie A . . . . .	37
6.2.4	Il problema del contenuto . . . . .	37
<b>7</b>	<b>La ricerca</b>	<b>38</b>
7.1	Ricerca interna . . . . .	38
7.1.1	Ricerca su misura . . . . .	39
7.1.1.1	Principio di Robert Browning . . . . .	39
7.2	Ricerca vincolata . . . . .	39
7.2.1	Ricerca vincolata dinamica . . . . .	40
7.2.2	Ricerca vincolata statica . . . . .	40
7.2.3	Soluzione ibrida . . . . .	40
7.3	Consigli . . . . .	40
7.4	Il search box . . . . .	41
7.5	Ricerche 2.0 (*) . . . . .	42
7.5.1	Il grande problema . . . . .	42
7.5.2	L'utente e le ricerche 2.0 . . . . .	42
7.5.3	Ivoluzione/evoluzione dell'interazione . . . . .	43
7.5.3.1	Il rumore . . . . .	43
<b>8</b>	<b>La visibilità</b>	<b>44</b>
8.1	Posizionamento SERP . . . . .	44
8.2	SPAMDEX (SEO) . . . . .	45
8.2.1	Parte testuale . . . . .	45
8.2.1.1	Term frequency (TF) . . . . .	45
8.2.1.2	Inverse document frequency . . . . .	45
8.2.1.3	Query SE . . . . .	46
8.2.2	Term spam . . . . .	46
8.2.2.1	Starter kit . . . . .	46
8.2.2.2	Quali keyword? . . . . .	47
8.2.3	Problema e soluzioni del term spamming . . . . .	47
8.2.3.1	Content hiding . . . . .	47
8.2.3.2	Redirection . . . . .	47
8.2.3.3	Cloaking . . . . .	48
8.2.4	Parte ipertestuale: pagerank . . . . .	48
8.2.4.1	Le catene di Markov e il random walk . . . . .	48

8.2.4.2	Problemi: spider traps e island . . . . .	49
8.2.5	Pagerank 2.0 . . . . .	49
8.2.5.1	Problema: dead ends . . . . .	49
8.2.6	Migliorare il pagerank . . . . .	50
8.2.6.1	Inlink . . . . .	50
8.2.6.2	Outlink . . . . .	50
8.2.6.3	Spam farm . . . . .	51
8.2.6.4	Contromisure SE . . . . .	53
8.2.7	Pagerank 3.0 . . . . .	55
8.2.7.1	Pagerank personalizzato . . . . .	55
8.2.7.2	Bontà di una pagina: doppio ranking . . . . .	56
8.2.7.3	Janus Graph . . . . .	56
8.2.7.4	Parte buona & parte cattiva . . . . .	56
8.2.7.5	Email with web . . . . .	57
8.2.7.6	Email 2.0: andiamo oltre . . . . .	58
<b>9</b>	<b>Il nome giusto</b>	<b>59</b>
9.0.1	Regole d'oro . . . . .	59
9.1	Indirizzi e nomi dal lato tecnico . . . . .	60
9.1.1	URI, URL e URN . . . . .	60
9.1.2	Struttura URI . . . . .	60
9.1.3	URI gerarchici . . . . .	61
9.1.4	URI opachi: URN . . . . .	61
9.1.5	Oltre gli URI . . . . .	61
9.1.6	Attacchi omografici . . . . .	61
9.1.7	Problema opacità URI . . . . .	62
9.1.7.1	L'idiozia umana . . . . .	62
<b>10</b>	<b>L'informazione e il web semantico</b>	<b>64</b>
10.1	La torre di Babele del web . . . . .	64
10.1.1	Un esempio di problema informativo . . . . .	64
10.1.2	Evoluzione . . . . .	64
10.2	Il web semantico . . . . .	65
10.2.1	Babel Fish: l'RDF . . . . .	65
10.2.1.1	Il modello RDF . . . . .	65
10.2.1.2	Funzionalità e proprietà di RDF . . . . .	66
10.2.1.3	I tipi semantici . . . . .	67
10.2.1.4	Caratteristiche di RDF Schema . . . . .	67
10.2.1.5	Oltre l'RDF Schema . . . . .	67
10.2.2	L'architettura del web (Tim Berners-Lee) . . . . .	67
10.2.2.1	Dare un nome ad ogni cosa: il problema degli URI . . . . .	68
10.2.2.2	La soluzione . . . . .	68
10.2.3	Strato ontologico: OWL . . . . .	68

10.2.3.1	Caratteristiche e proprietà di OWL . . . . .	68
10.2.3.2	Il problema: ragionarci . . . . .	69
10.2.4	SPARQL . . . . .	69
10.2.4.1	Modellazione con SPARQL . . . . .	70
10.2.5	Vocabolari ontologici . . . . .	70
10.2.5.1	DC: Dublin Core . . . . .	70
10.2.5.2	FOAF: Friend Of A Friend . . . . .	71
10.2.6	Un esempio . . . . .	71
10.2.7	Ancora SPARQL . . . . .	72
10.2.7.1	Extra SPARQL . . . . .	72
10.2.8	Un problema di complessità . . . . .	72
10.2.8.1	La soluzione di OWL . . . . .	73
10.2.8.2	Complessità e statistica . . . . .	74
10.3	Il web semantico diventa Linked Data . . . . .	74
10.3.1	Classificazione LOD . . . . .	74
10.3.2	Lifting & lowering . . . . .	75
10.3.3	Connettere le informazioni: il Mash Up . . . . .	76
10.3.3.1	Quanti punti scegliere? . . . . .	77
10.3.3.2	Esporre i dati . . . . .	77
<b>11</b>	<b>Mobile Web (e App)</b>	<b>79</b>
11.1	Un po' di storia . . . . .	79
11.2	Le App . . . . .	79
11.2.1	Parlano le statistiche . . . . .	80
11.2.2	L'arena delle App . . . . .	80
11.2.3	La sequenza della morte . . . . .	80
11.2.4	Alla ricerca dell'App . . . . .	81
11.2.5	ASO: App search optimization . . . . .	81
11.3	Usabilità: mobile e desktop . . . . .	81
11.3.1	L'esempio di Facebook . . . . .	82
11.3.2	Essere Mobile . . . . .	82
11.3.2.1	Timer su mobile . . . . .	82
11.3.2.2	Alla ricerca di soluzioni . . . . .	83
11.3.3	Taglia dello schermo . . . . .	83
11.3.3.1	Scroll verticale su mobile . . . . .	83
11.3.3.2	Invasività - pubblicità . . . . .	84
11.3.4	Interazione: le dita . . . . .	84
11.3.4.1	Fitts, il ritorno . . . . .	85
<b>12</b>	<b>Il social web (*)</b>	<b>87</b>
12.1	Storia . . . . .	87
12.2	Visibilità . . . . .	87
12.2.1	Regole virali . . . . .	87



<b>Appendices</b>	<b>88</b>
<b>A Analisi usabilità di un sito (*)</b>	<b>89</b>
A.1 Sito: note generiche . . . . .	89
A.2 Pagine: dettagli . . . . .	90

# List of Figures

5.1	Il comportamento degli utenti - Movimento utente e zone calde pagina . . . . .	29
5.2	Il comportamento degli utenti - Termografie . . . . .	29
5.3	Il comportamento degli utenti - Microsoft Office dopo e prima il restyling . . . . .	32
5.4	Il comportamento degli utenti - Evoluzione <i>scrollbar</i> su Mac .	33
5.5	Il comportamento degli utenti - Pie menu su <i>Second Life</i> . . .	34
7.1	La ricerca - Tipi di presentazioni risultati . . . . .	41
8.1	La visibilità - Esempi spam farm singole ottimali . . . . .	51
8.2	La visibilità - Spam farm alleanza profonda . . . . .	51
8.3	La visibilità - Spam farm alleanza superficiale . . . . .	52
8.4	La visibilità - Spam farm alleanza ring . . . . .	52
8.5	La visibilità - Spam farm alleanza complete core . . . . .	53
8.6	La visibilità - Forma ad alto livello del web . . . . .	54
8.7	La visibilità - Rappresentazioni grafiche dei link nel web . . .	54
9.1	Il nome giusto - Lenna e l'idiozia umana . . . . .	63
10.1	Il web semantico - <i>Layer</i> che compongono l'informazione . . .	65
10.2	Il web semantico - Grammatica di base . . . . .	66
10.3	L'informazione e il web semantico - Gerarchia dei problemi . .	73
10.4	Il web semantico - Informazione strutturata (sulla destra) rac- colta tramite <i>lifting</i> . . . . .	75
11.1	Mobile Web - Impugnature e utilizzo . . . . .	85

# Chapter 1

## Primi passi nel Web (\*)

1.1 1945 Memex

1.2 1960-68 NLS: onLine System

1.3 1960 Xanadus

Morale

“I sistemi sociali "Open World" devono essere gratis.”

1.4 1980 Enquire

1.5 1990 World Wide Web

1.6 Le alternative al WWW

## Chapter 2

# Il sito Web

### 2.1 La metafora del negozio

Possiamo considerare qualsiasi sito web come una casa o meglio un negozio: la gente guarda e poi decide se comprare o andarsene. L'**homepage** è la vetrina del negozio in cui le persone cercano informazione e questa informazione deve essere usufruibile nel modo più efficace possibile. A tal proposito sorge il problema di comunicare nei migliori dei modi l'informazione, un problema fortunatamente già affrontato dal giornalismo. Il pezzo informativo perfetto è il risultato dei 5 assi informativi principali, le così dette *5 W (6 W)*: Where - Who - Why - What - When (- How). Che nel web si traducono rispettivamente in:

**Where** A quale sito sono arrivato?

**Who** Chi c'è dietro questo sito?

**Why** Perché sono qui? Quale benefici mi dai?

**What** Che cosa mi offri? Mostramelo.

**When** Ultime novità del sito.

**How** Capito questo, come arrivo a quello di mio interesse?

### 2.2 Problemi e implicazioni

Il principale problema di un utente che visita il sito è il **TEMPO**. Bisogna sempre considerare che gli utenti hanno:

- aspettative.
- poco tempo, secondi contati!

Il sito quindi deve sapere offrire le  $6\ W$  nel pochissimo tempo che l'utente gli dedica. L'utente medio all'arrivo sulla homepage ha circa **31 secondi** prima di cominciare ad avere sensazioni negative. 31 secondi. Solo 31 secondi per convincere l'utente. Questo porta ad una serie di implicazioni:

### 2.2.1 Quanto testo nella homepage?

Un uomo adulto di buona cultura legge dalle 200 alle 300 parole al minuto, su computer però la velocità di lettura è più bassa: 180 parole al minuto. Con più di 93 parole abbiamo già superato il limite di 31 secondi, se teniamo poi conto dell'intero layout allora sono decisamente troppe.

### 2.2.2 Il comportamento dell'utente è dinamico

Bisogna far sì che l'utente al nostro sito ci ritorni ma ora le  $W$  di Who, Where e Why non sono più richieste. Fortunatamente l'utente salta alcuni pezzi ma ora ha ancora meno tempo da dedicare.

### 2.2.3 Questione di tempo

Di seguito i tempi medi di permanenza:

- 1<sup>a</sup> volta: 31 secondi.
- 2<sup>a</sup> volta: 25 secondi.
- 3<sup>a</sup> volta: 22 secondi.
- 4<sup>a</sup> volta: 19 secondi.
- dalla 5<sup>a</sup> volta in poi i tempi sono stabili.

Dalla seconda volta in poi quello è il patrimonio dei secondi da dedicare agli assi What, When e How, corrispondono a 57 parole al massimo! Una home prolissa non darà mai tutti gli assi informativi nei pochi secondi a disposizione e una home poco chiara (assi informativi mancanti) darà un motivo in più per scappare all'utente.

### 2.2.4 E il resto del sito?

Per tutte le altre pagine non abbiamo bisogno che gli assi siano il principale obiettivo informativo. Inoltre l'utente una volta superata la homepage (vetrina) dedica più tempo. Dai 31 secondi passa a **53 secondi** che corrispondono a circa 160 parole in cui includere info più specifiche. Un ottimo modo per gestire il poco numero di parole è quello di attirare l'attenzione dell'utente con descrizioni corte che conducano ad altre pagine per ulteriori informazioni. Ciò fa sì che solo l'utente effettivamente interessato leggerà il

testo più lungo mentre agli altri verrà fatto perdere meno tempo. Sembrerebbe un'ottima soluzione quella di spezzare la pagina e resettare i timer guadagnando tempo ma attenzione perché oltre al tempo singolo di ogni pagina bisogna considerare anche il **tempo globale**.

### 2.2.5 Questione di tempo - Parte II

Il **tempo globale** rappresenta il tempo massimo dell'utente per raggiungere lo scopo. Si suddivide in due:

**Tempo preliminare:** è il tempo che un utente dedica per convincersi a restare nel sito, per questo chiamato anche **tempo di scelta**. Il tempo di scelta medio è di 1 minuto e 50 secondi, allo scadere di questo timer l'utente abbandona il sito indipendentemente se esso conteneva l'informazione ricercata o no. Nell'88% dei casi quell'utente non ritornerà più.

**tempo complessivo:** l'utente è convinto a restare per cui dedica fino a 3 minuti e 49 secondi per avere successo altrimenti abbandona.

#### Morale:

"È molto importante il bilanciamento tra homepage e pagine interne"

Al primo accesso infatti l'utente dopo aver navigato homepage e una pagina interna decide se restare o andarsene (1:50 - tempo di scelta). Dopo tre pagine e mezzo l'utente deve aver successo in quello che doveva fare.

## 2.3 L'importanza della struttura

Abbiamo capito:

- dopo un click l'utente deve essere convinto a restare;
- dopo tre link l'utente deve avere quello che cercava.

Ora nelle pagine interne che assi informativi servono? Sembrerebbe che non serva replicare le info della homepage nelle pagine interne ma la navigazione al giorno d'oggi non attraversa quasi mai la homepage! Grazie ai motori di ricerca infatti la navigazione comincia da qualunque punto di qualsiasi sito.

### 2.3.1 Deep linking

Questo fenomeno viene chiamato **deep linking** ovvero avere il link interno di un sito. Accade questo perché anche i motori di ricerca hanno i loro timer e devono dare nel modo più diretto l'informazione giusta che l'utente cerca. Ogni pagina quindi può essere una pagina iniziale per un utente. La metafora del negozio si fa critica perché ciò significherebbe avere clienti teletrasportati all'interno, già tra gli scaffali.

### 2.3.2 Gli assi in dettaglio

Andiamo quindi a vedere gli assi in dettaglio e come bisogna gestirli in seguito al *deep linking*. Alcuni assi risultano essere **obbligatori** per tutte le pagine:

- Who: il logo (solitamente da preferire in alto a sinistra).
- What: tipicamente un link alla homepage.

Altri risultano essere **opzionali**:

- When: le novità del sito.

Altri ancora entrano nella categoria **opzionali consigliati**:

- Why: basta una breve descrizione, uno slogan.
- How: funzionalità di search (da preferire in alto a sinistra).

### 2.3.3 L'importanza del Where

Un paragrafo a parte invece è doveroso dedicarlo all'asse Where. Infatti in ogni pagina si dovrebbe rendere chiaro il contesto in cui l'utente si trova. Si potrebbe obiettare con "perché non mandarlo alla homepage" come per l'asse What ma ciò costituirebbe un link in più all'utente e, peggio, spostare l'utente dal luogo in cui c'è l'informazione di suo interesse. Conviene dargli informazioni del where nella pagina interna. Per fare questo si utilizza il *breadcrumb*, ne esistono di tre tipi:

**Location:** dà il posto della pagina nella gerarchia del sito. Ad esempio: "Home » Categoria » Pubblicità » Pagina".

**Attribute:** mostra la categoria e gli attributi della pagina. Un po' come gli hashtag odierni. Un pagina può trovarsi sotto più categorie.

**Path:** mostrano il cammino dell'utente per giungere alla pagine. È dinamico infatti dipende dal cammino dell'utente e usa dei *cookie* per tenere traccia di tali informazioni.

### Pro e contro

- **Path** non risolve il problema del Where dopo che l'utente è catapultato nella pagina.
- **Attribute** sembra la scelta migliore ma implica un sistema più complesso per gestire il sito e raggiunge taglie troppo grandi in certi casi.
- **Path** resta una soluzione semplice e lineare.

**Separatori** Per completezza si riportano i separatori per *breadcrumb* più comuni:

- segno di maggiore ">"
- segno di doppio maggiore "»"
- backslash "\"



## Chapter 3

# Problemi di usabilità

### 3.1 Problemi persistenti

Sono i problemi gravi fortemente connessi alla tecnologia e che nel tempo non cambieranno.

#### 3.1.1 Navigazione

Il problema del navigare nel web odierno è la possibilità di perdersi: *lost in navigation*, ossia la prese di coscienza dell'utente che capisce di essersi perso. Fortunatamente, se opportunamente inserito, l'asse informativo Where risolve questo tipo di problema.

##### **Morale:**

“Gli utenti devono essere coscienti di dove sono e dove devono andare.”

##### **3.1.1.1 Dove sono, dove ero e dove sarò**

Nonostante l'uso di *breadcrumb* e asse Where opportunamente comunicato all'utente ciò non basta. Può infatti capitare che l'utente si ritrovi in pagine già visitate e deva ricordarsi i percorsi già fatti. Lo sforzo diventa pesante e crea malumore. Per non far affaticare l'utente esiste al giorno d'oggi una consuetudine non standard riconosciuta in tutto il web che è quella di colorare diversamente i colori dei link già visitati. Ciò fu implementato da Netscape Navigator e da allora è diventata una buona norma per garantire maggior usabilità. Il 75% dei siti web usa il cambio colore dei link già visitati.

##### **Morale:**

“All'utente pesa meno la grafica rispetto alla funzionalità e allo sforzo.”

### 3.1.2 I movimenti dell'utente

Le azioni generali per interagire che può utilizzare l'utente sono:

- il *click*.
- il *back* (pulsante prezioso, presto vedremo il perché).

Secondo gli studi sul comportamento degli utenti sul web si è scoperto che ad essi piace navigare all'indietro, anzi lo adorano. Prendiamo ad esempio la visita di un sito in cui si sia andati in profondità di 4 livelli e si deva tornare alla homepage. Gli utenti a questo punto spesso invece di cliccare una volta il link diretto (magari sul logo del sito) preferiscono di gran lunga utilizzare il pulsante *back* ripetutamente. Si arriva fino a 7 click del pulsante *back* anche in presenza di un link diretto. È lo stesso comportamento che si tiene con il telecomando della propria TV. A volte basterebbe premere i pulsanti numerici per passare ad un diverso canale ma si preferisce spostarsi di un canale alla volta usando un unico bottone invece di due o più bottoni numerici. Questo uso comune è noto come *backtracking*.

#### Morale:

“La pulsione primaria dell'utente non è quella di minimizzare il tempo ma quella di minimizzare lo sforzo.”

L'uomo ha orrore nello sforzo previsto nel futuro e tende a fare cose folli e illogiche per allontanare tale sforzo (si veda *l'algoritmo della carta igienica*). Quindi, gli utenti minimizzano lo sforzo computazionale e per fare ciò ricorrono all'uso del pulsante *back* perché:

- non serve ricordarsi il percorso;
- non bisogna trovare il tasto *back*, (è sempre lì garantito).

Da ciò ovviamente ne consegue che non bisogna **mai togliere l'uso del *back button***

### 3.1.3 Nuova finestra? No, grazie

Un altro problema persistente è quello di aprire una nuova finestra di navigazione anziché usare sempre la stessa. Esistono due tipi di finestre, il tab e la nuova finestra vera e propria. L'aprire una nuova finestra ha gravi conseguenze per l'utente medio:

- Non c'è più la cronologia di navigazione (addio *back button*!)
- Avere finestre diverse aperte confonde l'utente medio.

Analizziamo nel dettaglio che cosa succede all'apertura di una nuova finestra. Prima di tutto questa si sovrappone alla navigazione esistente provocando panico per l'utente medio. Se dovesse non sovrapporsi l'utente medio seleziona quella bassa lasciando l'altra finestra aperta. Di conseguenza il link della pagina già aperta non funziona più perché la pagina risulta aperta ma di ciò l'utente medio non ne è a conoscenza.

#### 3.1.3.1 Un problema correlato: i pop-up

Un problema che si collega molto con l'apertura di una nuova finestra è quello dei pop-up. Piccole finestre che si aprono senza il permesso dell'utente (si veda in seguito per maggiori dettagli).

#### 3.1.4 Convenzioni violate

Le convenzioni non sono standard ma semplicemente la prassi, ciò che fanno tutti e per questo più familiari all'utente.

#### Legge di Jacob

"Gli utenti spendono la maggior parte del tempo su altri siti web."

Gli utenti sono abituati a navigare in altri siti quindi non abbiamo il potere di fare tutto di testa nostra solo perché è il "nostro" sito.

#### 3.1.5 Altri problemi: What non rispettato

Mai usare linguaggio vuoto o con poco contenuto/slogan. L'utente che visita una pagina si aspetta contenuto non "politichese" cit.

**Problema correlato: la forma del testo** Il contenuto di una pagina web conta ma il testo deve sempre avere una forma semplice, chiara e sintetica; la lettura su schermo è diversa dalla normale lettura su carta. Mai usare testo difficile e monolitico che spesso, purtroppo, è usato nei siti delle pubbliche amministrazioni. Il testo usato su altri media non è adatto al web. Alcuni accorgimenti per evitare ciò è quello di tagliare testo.

- Se abbiamo del normale testo da inserire in una pagina bisogna **dimezzare** per far sì che diventi testo web.
- Se abbiamo testo generico, il testo web è circa **un quarto**.

Un altro suggerimento per scrivere testo adatto al web è quello di cominciare con la conclusione e successivamente espandere.

## 3.2 Problemi non-persistenti / Il contenuto

### 3.2.1 Splash page

Le *splash page* Sono le pagine di presentazione che sostituiscono la homepage. Evitarle a tutti i costi, fanno perdere tempo all'utente soprattutto se sono anche animate. Molto meglio una homepage semplice che comunica in modo adeguato i 6 assi principali.

### 3.2.2 Richieste di registrazione

Altra cosa da evitare: mai richiedere informazioni personali all'utente, soprattutto mai richiedere una registrazione prematura. Su 10 utenti appena l'1,1 è disposto a dare la propria mail. Bisogna infatti tenere conto di:

- l'utente deve sapere se vale la pena o no (problema di *trust*).
- La registrazione richiede sforzo computazionale (nuova login e password!).

Alcuni siti arrivano pure a bloccare la prima visita con un pop-up richiedendo l'iscrizione. Come può un utente in questo modo capire se fidarsi o se non li è lasciata opportunità di visitare prima il sito? Ogni richiesta di registrazione deve avvenire dopo aver convinto l'utente.

### 3.2.3 Lo scrolling maledetto

Parliamo di *scroll* verticale. I dati mostrano che in media gli utenti "*scollano*" 1.3 schermi. Questo significa che in totale la parte visualizzata di una pagina corrisponde a 2.3 schermi. Tutto quello che viene dopo (in media) non viene visto. Per cui, attenzione alla struttura del layout della pagina e alla posizione dei contenuti.

- Nella homepage solo il 23% effettua lo *scroll*.
- Nelle pagine interne il 42%.
- Visite ripetute alla homepage riducono l'uso dello *scroll* al 14%.

Riportiamo l'esempio da non imitare dell'attuale (07/02/2016) pagina di presentazione dell'iPod nano: <http://www.apple.com/it/ipod-nano/>. Alcune osservazioni:

- All'apertura la figura in primo piano è tagliata (potrebbe essere un modo per incoraggiare lo scroll).
- Il testo anche se conciso non dice nulla di utile all'utente.
- La pagina è composta da 14 (!!!) *scroll*.

### 3.2.3.1 Taglia dello schermo

Un gran gratta capo di oggi per i siti web è la taglia dello schermo (risoluzione). Ad oggi sono numerosissime. Negli anni passati 1024x768 era una taglia di riferimento ma con l'avvenuta dei netbook (1014x600 massima) il trend è cambiato. Inoltre non è detto che tutti massimizzino la finestra per cui statisticamente la taglia più sicura su cui affidarsi è la 800x600. Con il mobile le cose peggiorano. Non conta più la taglia dei pixel (risoluzione) ma dello schermo vero e proprio. Esistono infatti piccoli schermi con risoluzioni alte. Per risolvere questo problema troppo spesso si è ricorso al *frozen layout* ossia fissare il design per una taglia con il risultato di avere effetti disastrosi con le taglie più grandi. Se si fissa l'asse orizzontale otterremo infatti con uno schermo grande una pagina piccola e contenuta con ovvio spreco di spazio con un schermo piccolo invece l'odiato *scroll* orizzontale.

### 3.2.3.2 Scrolling orizzontale

Lo *scroll* orizzontale è odiato dagli utenti ed è molto peggio del verticale perché:

- non è comune
- e non rispetta la normativa classica del testo.

Nella lettura l'asse delle ascisse è fissato mentre viene effettuato lo *scroll* sull'asse delle ordinate con gli occhi. Inoltre avere entrambi gli *scroll* porta a dover gestire uno spazio di 2 dimensioni con logica conseguenza di attivare più sforzo computazionale.

### 3.2.3.3 People do scroll

Potrebbe essere interessante approfondire la questione dello *scroll*. Da un lato abbiamo visto che lo *scroll* è uno sforzo in più richiesto all'utente ma il tempo passa e il comportamento e le abitudini degli utenti possono cambiare (soprattutto il boom mobile). Ci sono molti studi che indicano che gli utenti sono pronti a fare lo sforzo di *scroll* se il layout lo incoraggia. Ulteriori approfondimenti <http://it.uxmyths.com/post/28647124262/mito-3-le-persone-non-scrollano>.

## 3.2.4 Lo sforzo computazionale spiegato da Engelbart

Abbiamo parlato nella sezione La storia del Web (<https://www.youtube.com/watch?v=1MPJZ6M52dI>) della straordinaria invenzione di Douglas Engelbart dove oltre ad il primo mouse della storia si vedeva una tastiera da 5 tasti. Essa permetteva di memorizzare fino a 31 combinazioni di tasti associate ad un evento sulla macchina. Questa tecnologia non è sopravvissuta proprio per il troppo sforzo computazionale richiesto. Per questo motivo per

ogni cosa chiedetevi sempre qual è lo sforzo che un utente deve compiere e cercate di minimizzarlo, l'uomo si muove sempre verso quella direzione.

### 3.2.5 Bloated design

Il *bloated design*, letteralmente design gonfiato, è un altro tipico errore che abita il web odierno. Il *bloated design* si ha quando il sito presenta troppi effetti, questo risulta essere **statisticamente fastidioso** poiché aumenta lo sforzo computazionale. Nella storia del web questo si è presentato con la lotta tra i *browser* dove si creavano comandi con effetti bizzarri e inutili per l'utenza. Alcune anni dopo un comando di questi, il *blink tag* fu definito dallo stesso autore come "la cosa peggiore per internet". Di esempi di *bloated design* ce ne sono centinaia:

- uso di musica con avvio automatico al caricamento.
- effetti sconvolgenti che confondono l'utente.
- siti di design in cui risulta complicata la navigazione.
- e altro ancora...

### 3.2.6 Abusi multimediali

#### 3.2.6.1 Il 3D - Prima, dopo, ora

Perché l'interfaccia 3D non è entrata nel mondo del web? Già nel 1922 fu proposto nella televisione ma non ebbe successo. Ancora una volta il limite dell'umano, la necessità di minimizzare lo sforzo computazionale determina il fallimento di una tecnologia proprio come la tastiera di Douglas Engelbart. Non limitiamoci solo al web, perché non rendere l'interfaccia dei sistemi operativi in 3 dimensioni? Ci ha provato Anand Agarawala nel 2007, designer, ([https://www.ted.com/talks/anand\\_agarawala\\_demos\\_his\\_bumptop\\_desktop](https://www.ted.com/talks/anand_agarawala_demos_his_bumptop_desktop)) proponendo una interfaccia virtuale che simula la fisica in 3 dimensioni. Perché questa interfaccia non è ancora nei nostri dispositivi? La difficile interazione con esso si è rilevata più importante che ne la bellezza visiva. Se possiamo evitare l'uso smoderato della multimedialità, il sito commerciale J. Crew lo sapeva bene quando per mostrare i propri vestiti non ha usato nessuno effetto. Una scelta banale ma è quello che l'utente vuole.

#### Morale:

"Conviene offrire *snapshot* 2D di oggetti 3D con complessità bassa"

### 3.2.6.2 I plugin

Una nota per i *plugin*: soffrono di un problema fondamentale, non sono standard e richiedono un'installazione quindi sforzo aggiuntivo. Il comportamento dell'utente di fronte alla richiesta dell'installazione di un *plugin*:

- “non so cosa può succedere quindi non lo faccio” (vedi gli aggiornamenti di Windows 10 ora nascosti all'utente).
- Installare un *plugin* fa perdere tempo (una richiesta di installazione di plugin fa perdere il 90% degli utenti non fidelizzati).

### 3.2.6.3 Dai plugin a Flash!

Si potrebbe pensare che con *Flash* i problemi dei *plugin* non si hanno più. Sbagliato!

- *Flash* è sempre un *plugin* con necessità costante di aggiornamenti.
- Tempo di caricamento aumentato.
- Dà molti mezzi e più libertà espressiva, un vantaggio che diventa problema se si cade nel già visto *bloated design*.

Evitare *Flash* non significa evitare questi problemi. Anche con il recente HTML5 si può cadere in trappole come il *bloated design*. Tutto dipende dall'uso.

### 3.2.6.4 I video

Un altro strumento multimediale è l'uso dei video, oggi sempre più in espansione (si pensi a Facebook e al recentissimo Snapchat). Il principale vantaggio è lo stesso della televisione: basso costo computazionale. Di contro abbiamo la richiesta di più risorse (banda) e il timer collegato alla durata del video.

- Tempo medio consigliato: 1 minuto.
- Tempo massimo consigliato: 2 minuti.

Questi sono consigli generali ma molto dipende dal **target** che si ha, ad esempio youtube non ha questi limiti.

### 3.2.7 La Metafora visiva

Altro problema che nel web produce disastri sull'usabilità dei siti web. La metafora visiva si ha quando l'utente è ingannato dall'aspetto grafico che dà aspettative illusorie e le tradisce. Per esempio:

- Il pulsante dello *scroll* sostituito da un immagine.
- L'immagine con scritto "clicca" non cliccabile.
- Un testo in grassetto colorato che ricorda un link ma non lo è.
- Pulsanti non cliccabili.
- Immagini non riconoscibili come immagini.
- Pulsanti che sono parzialmente cliccabili.
- ...

Le metafore visive tradite non hanno soltanto a che fare con link, pulsanti e immagini ma anche con **concetti**. Ad esempio dei concetti che significano qualcosa ma non sono intuibili subito.

### 3.2.8 I menu di navigazione

Per quanto riguarda i menu dei siti internet abbiamo un vantaggio e uno svantaggio.

- (vantaggio) Essi sono usati già nei sistemi operativi per cui gli utenti hanno **familiarità** con essi.
- (svantaggio) I menu del desktop sono costituiti da comandi, i menu del web di informazioni e quando c'è troppa informazione i menu "esplodono" in dimensione.

Nell'interazione con il mouse a questi menu molto grandi abbiamo degli effetti disastrosi sull'usabilità.

- L'83% degli utenti non c'entra la casella al primo colpo.
- Il 53% esce con il mouse e chiude il menu.

#### 3.2.8.1 Pathfinding

Per capire queste percentuali analizziamo come gli utenti si muovono nelle pagine web, il così detto *pathfinding*. L'algoritmo umano di spostamento da A a B è una linea retta, sforzo computazionale minimizzato. Nei menu con sotto-menu (**menu a cascata**) questo non funziona. Di fronte a questi tipi di menu il 92% degli utenti esce dal menu con conseguente chiusura di esso. Per questo motivo i **menu a cascata** devono avere al massimo due livelli ed essere *fault-tolerant* cosicché all'uscita del menu esso non si chiuda immediatamente creando frustrazione all'utente.



### 3.2.9 Il testo

Torniamo a parlare del testo dopo aver visto che:

- il testo per il web è diverso dal testo normale,
- deve essere semplice e sintetico
- e non deve essere scritto in "politichese".

Vediamo ora altre regole che possono generare sensazioni negative se non rispettata.

1. Il testo deve **essere leggibile** con **dimensioni adatte**. La grandezza minima è di 10pt per ogni schermo per carattere.
2. Rinunciateci. A qualcuno non andrà bene il vostro testo. Sempre. Per andare incontro a questi mettere a disposizione delle opzioni visibili per il *resize*. (Questo infatti è divenuto uno strumento *zoom* dei browser)
3. Il testo è testo e deve essere riconosciuto come tale. Non usarlo mai come grafica con cambi di *font*. Sempre avere un solo *font* (al massimo 2) consistente per tutta la pagina, meglio per il sito. Il *font* a cui gli utenti sono più abituati risulta **Verdana**.
4. Fare sempre attenzione al contrasto tra colore del testo e gli ulteriori sfondi delle pagine.

#### 3.2.9.1 Caps lock

Per quanto riguarda il *caps lock* dai dati è risultato più difficile da leggere del 10%, principalmente perché non si è abituati.

#### 3.2.9.2 Immagini sostitutive

Per le immagini invece che sostituiscono il testo abbiamo conseguenze molto negative:

- L'immagine **non scala** e aumenta il carico della pagina.
- Non permette l'uso dello strumento *past & copy*.
- Non interagisce con il motore di ricerca che lo considera immagine appunto.

### 3.2.9.3 La maledizione Lorem Ipsum

Il *lorem ipsum* è una tecnica molto in voga nel web design. Permette infatti di strutturare un layout senza la presenza del testo effettivo che andrà inserito come contenuto. *Lorem ipsum* infatti è un testo latino senza significato generato automaticamente che va a riempire le parti di testo di una pagina solo ai fini del layout. Facendo così il layout visivo si stacca dal testo dando più priorità al layout quando invece è il testo la cosa più importante. Usare il *lorem ipsum* molte volte crea siti con scarsa usabilità. Il testo non è da considerare come un elemento di blocco per garantire l'usabilità ma deve essere strutturato secondo certe logiche (vedi Scanning).

**L'effetto ghigliottina** Usando il *lorem ipsum* di fatto otterremo alcuni casi dove il testo sfiora il layout quando inseriamo il vero testo. Oppure addirittura si ricorre ad uno *scroll* interno alla pagina perché lo spazio previsto risulta insufficiente.

### 3.2.9.4 Scanning

Cosa fanno gli utenti appena arrivano in una pagina web? Avviene lo *scanning* veloce della pagina non per forza ordinato. Questo fa sì che l'utente si crei un'immagine mentale approssimata di come è fatta la pagina per minimizzare lo sforzo. Sarà poi l'input continuo dell'occhio a dare una mappa visiva informativa fatta meglio. Per questo motivo i grossi blocchi di testo richiedono troppo sforzo e durante la fase di *scanning* il contenuto di essi non è percepito. Il testo deve avere una struttura propria.

### 3.2.9.5 Strutturazione

Per strutturare bene il testo occorre:

1. Dividere il testo in blocchi.
2. Dare un titolo descrittivo ad ogni area del testo.
3. Usare liste *itemize*.

Questi accorgimenti aiutano la prima organizzazione/*scanning* della pagina. Il titolo (*keyword*) del blocco dovrebbe essere breve e pertinente. Si possono usare parole evidenziate (uso di *bold*) che aiutano lo *scanning* della pagina, tenendo conto che un utente memorizza fino a 6/7 *keyword*. I link tendono ad essere più evidenti delle *keyword* sono assorbiti dallo *scan* iniziale. Di seguito elenchiamo alcuni accorgimenti per i link:

- I link non devono essere troppo lunghi.
- I link non devono avere nomi troppo simili per aiutare lo *scanning*.

- I link non devono essere mai del tipo "clicca qui", questo significa dare allo *scanning* la *keyword* "clicca qui" che non ha nessun significato, confonde l'utente e aumenta lo sforzo computazionale richiesto.

Le liste (terzo punto) aumentano la chiarezza e il grado di soddisfazione dell'utente (+47%) si usano con almeno quattro elementi perché con meno affatico l'utente. Attenzione però! L'efficienza delle liste **decresce linearmente** con il numero di liste disposte verticalmente, mentre **decresce esponenzialmente** con le lista disposte sull'asse orizzontale.

#### 3.2.9.6 Blonde effect

Il *blonde effect* avviene quando si ha una percezione errata a causa dei limiti dell'utente. Durante lo *scan* l'utente può percepire erroneamente alcune zone, bisogna quindi sempre porre attenzione a come lo *scanning* agisce sulle nostre pagine. Per capire perché "blonde" <https://www.youtube.com/watch?v=DctVteQDRIM>

## Chapter 4

# Usabilità: Siti E-Commerce

Di un sito commerciale, con scopo la vendita di prodotti, si pensa che la cosa più importante sia appunto il prodotto. In verità oltre al prodotto è molto importante anche il prezzo! Entrambi sono fondamentali in un sito E-Commerce.

### 4.1 Il prezzo

Gli utenti vogliono il prezzo nel modo più semplice possibile ovvero vicino al prodotto. Purtroppo in molti casi questo non avviene richiedendo all'utente sforzo che genera frustrazione e spesso avviene l'*iper-associazione*.

#### 4.1.1 Trappola iper-associazione

È un problema che si verifica quando il prezzo è tenuto nascosto fino all'ultimo link del prodotto. Questo genera conseguenze negative.

- Ulteriore fatica computazionale, per sapere il prezzo è necessario un passaggio in più.
- Si perde il beneficio della mappa creata con lo *scanning*.
- *Gambling click*: l'utente è costretto a cliccare col dubbio di fare qualcosa di giusto. Si è misurato che in presenza di un *gambling click* il gradimento del sito scende del 40% e il link è cliccato solo dal 30% in media.

Se non conosco i prezzi? Dobbiamo dare quantomeno un *range* approssimato di prezzo qualsiasi, l'importante è che ci sia un prezzo. Nel web non si deve mai mettere alla prova la fiducia di un utente.

## 4.2 La pubblicità

### 4.2.1 Pubblicità classica e tecniche

Nelle pubblicità classiche che troviamo oramai dappertutto lo scopo è impressionare e colpire l'utente per poco tempo. A tal scopo sono state inventate alcune tecniche:

**Fishing price:** utilizzo di un prezzo esca che non è quello reale.

**Net price:** utilizzo di un prezzo netto e quindi inferiore a quello che l'utente pagherà

Nel web si tende a comparare la pubblicità in un sito web come la pubblicità nella realtà e quindi si riusano le tecniche classiche sopra descritte. Vedremo che non è così, il web è un luogo diverso. Ricordiamo la metafora, quando si è in un sito internet è come essere all'interno di un negozio fisico.

### 4.2.2 Noi e la pubblicità

Il nostro cervello è dotato di una memoria a breve termine e una a lungo termine. Il flusso di dati proveniente dalla pubblicità classica è principalmente memorizzato in quella a breve termine, solo una piccola parte è salvata in quella a lungo termine. Quest'ultime informazioni sono il vero e proprio succo della pubblicità, un'idea vaga senza dettagli. Proprio questa sensazione vaga induce l'utente a venire in negozio o a comprare quel prodotto piuttosto di quell'altro.

### 4.2.3 La pubblicità nel web

Nel web non si può ricorrere all'idea vaga nella memoria a breve termine, si è lì nel sito, il prezzo è lì segnato e si ricorda. Non è possibile alterare quindi il prezzo per invogliare poi all'acquisto, il periodo temporale è troppo breve. Utilizzare i trucchi della normale pubblicità quindi porta ad innervosire l'utente.

**Fishing prize nel web:** 9 utenti su 10 abbandonano il sito, l'1 che resta ha un calo di *trust* del 50% (meno gradimento e timer ridotti).

**Net price nel web:** l'85% degli utenti abbandona il sito.

Un esempio di *net price* è quello di non includere le spese di trasporto o assicurazione solo alla fine, l'utente vedrà i soldi aumentare appena prima dell'acquisto creando insoddisfazione e portandolo addirittura ad annullare l'acquisto. La soluzione è quella di usare un carrello dove mostrare ben in vista le spese senza però richiedere i dati personali fino all'effettivo pagamento. Si ricorda infine la parola magica **gratis** o *free* che è un vero e proprio attivatore di sensazioni positive.

## 4.3 Il prodotto

### 4.3.1 Descrizione del prodotto

È sempre richiesta e deve essere in un linguaggio il più possibile semplice e chiaro in modo che sia comprensibile per tutti gli utenti. Mai assumere che l'utente sappia cos'è il prodotto e gli interessi soltanto il prezzo, l'utente si aspetta sempre una descrizione completa altrimenti il sito dà l'impressione di non essere professionale e questo può portare in taluni casi all'abbandono del sito. Oltre alla completezza se le descrizioni sono mal fatte queste portano alla migrazione dell'utente nel 99% dei casi anche dove il prezzo sia più basso gli utenti sceglieranno quello più caro. Se il nostro prezzo è il 20% in meno solo il 5% degli utenti ritornerà a comprare nel nostro sito. La descrizione va fatta con cura da essa l'utente basa l'acquisto per una questione di *trust*, il prezzo competitivo quindi non basta.

### 4.3.2 L'aspetto visivo del prodotto

Ricordarsi sempre: l'utente vuole vedere il prodotto e richiede sempre una descrizione visiva (immagine) con qualità full-screen. Un'immagine di un prodotto deve essere sempre cliccabile in modo da mostrare più nel dettaglio il prodotto. La visualizzazione nel dettaglio deve essere ad esclusiva scelta dell'utente altrimenti spesso l'utente non è disposto ad aspettare il caricamento. Ricordiamo inoltre che le immagini 2D semplici sono più apprezzate dagli utenti.

Riportiamo il sito di [www.jcrew.com](http://www.jcrew.com) come esempio di sito E-Commerce. Il sito implementa molte caratteristiche spiegate egregiamente mentre in altre pecca clamorosamente.

## Chapter 5

# Il comportamento degli utenti

Nel 1990 e 1991 si sono effettuati diversi studi sul comportamento degli utenti con metodologie varie. Da questi è sorto che per ogni tipo di media l'uomo mette in atto delle regole ben precise. Vedremo un esempio di media classico e il perché le sue regole **non** sono da riutilizzare nel web, quali sono le differenze e quali strumenti abbiamo a disposizione per diminuire gli sforzi dell'utenza.

### 5.1 Un media classico: Il giornale

Nel giornale la principale fonte di attrazione è costituita dalle immagini, dalle foto. Più colorate sono più per lo *scanning* quel blocco è fonte di informazione.

#### 5.1.1 L'importanza delle immagini

Sui giornali le immagini vincono sul testo 80 a 20. I pezzi di testo accompagnati da immagini infatti risultano quelli più letti. **L'immagine più grande** è il primo **punto d'entrata** e d'attrazione primaria per invogliare l'utente alla lettura. Inoltre in un giornale due pagine aperte sono percepite come un'unica grande pagina.

### 5.2 Il web

Alla luce di queste informazioni del comportamento tenuto dagli utenti con i giornali si può forse pensare che per le pagine web sia lo stesso. Non è così! La modalità di lettura differisce ma non solo, anche lo spostamento dell'attenzione ha un comportamento diverso come vediamo in figura [5.1](#).



Figure 5.1: Il comportamento degli utenti - Movimento dell'utente e zone "calde" di una pagina

### 5.2.1 Fasce d'attenzione

Per misurare il movimento fatto da utenti in una pagina web si usa la tecnica dell'*eyetracking* da cui si ricava una termografia di una pagina dove le zone calde corrispondono alle dove l'utente dedica più tempo mentre quelle più fredde quelle che non vede affatto se non nella prima fase di *scanning*. Le diverse termografie classiche indicano che il **punto d'entrata** di una pagina web è **in alto a sinistra**. Le fasce d'attenzione risultano quindi avere una forma a *F* o a *cono gelato* come si può notare nella figura 5.2. Nella maggioranza dei casi per ogni schermata della stessa pagina abbiamo una forma a *F* nel punto in cui avviene lo *scroll* si ha il *blind spot* un punto cieco che non sarà mai visualizzato.



Figure 5.2: Il comportamento degli utenti - Esempi di termografie



### 5.2.2 L'importanza del testo

Al contrario dei giornali il testo è molto più rilevante rispetto alle immagini nel web. Per sfruttare questo ecco alcuni consigli per migliorare una pagina web.

- Il punto d'attrazione (in alto a sinistra) necessita di testo, il logo deve essere sempre accompagnato da qualche scritta magari che risponda all'asse What.
- Il testo su una singola colonna è meglio che su più colonne come nei giornali (l'effetto è molto simile alle liste *itemize* orizzontali).
- Le *keyword* messe vicine diluiscono l'importanza data nello *scan*.
- Il *bold* non basta per evidenziare le parole chiave. Meglio usare anche la sottolineatura (come un hyperlink) e ingrandire il *font* o meglio ancora separare la o le *keyword* in una riga a parte (una sorta di mini titolo).

#### 5.2.2.1 Paragrafi e titoli

Un altro utile accorgimento per sfruttare l'importanza del testo è quello di dividere i grandi blocchi di testo in paragrafi corti (rispetto a quelli lunghi attraggono alla lettura quasi il doppio). Lo stesso possiamo dire dei titoli, più sono corti meglio è. Inoltre più un testo è spezzato in piccoli paragrafi più gli utenti sono incoraggiati a leggerlo rilassando i timer (suggerimento valido anche nelle mail). Sfruttare il *blurp*, cioè aggiungere ad ogni titolo una mini descrizione per migliorare l'attrazione dell'intera pagina. I *blurp* offrono vantaggi:

- rilassano i timer, ma non cambiano la voglia di proseguire nella navigazione del sito;
- aumentano il tasso di ritorno della pagina.

Il contenuto del *blurp* non è un blocco unico ma può essere suddiviso in parti per cui ha una sua struttura e una sua termografia, per questo motivo le parole fondamentali devono stare alla sinistra del *blurp*.

### 5.2.3 Pagine grasse o magre?

Per pagina grassa si intende una pagina web con contenuto più spazioso mentre con la pagina snella una pagina con contenuto più compatto. Dalle analisi è emerso che la pagina grassa funziona peggio. Bisogna infatti porre molta attenzione al separare il testo in paragrafi perché si può incorrere nel *diluted design*. La spaziatura e la separazione portano ad uno *scanning* più veloce ma ciò funziona male nelle pagine ricche di contenuto testuale. In generale:

- la **pagina grassa** è da preferire in quelle pagine di navigazione con link e poco *scroll*;
- la **pagina magra** invece è da preferire in quelle pagine con molto contenuto e tanto *scroll*.

#### 5.2.4 Immagini nel web

Abbiamo visto che le immagini nel web hanno un ruolo marginale rispetto al testo difatti un'immagine affiancata ad un testo perde quasi tutta l'attenzione. Nonostante questo hanno comunque una proprietà (diversa dal testo) che le rende utili se sfruttata al meglio. Innanzitutto fissiamo una taglia minima: 210x230 pixel, più piccola sembra un'icona e c'è rischio di ingannare l'utente. Le immagini hanno un tasso di click superiore al testo, solitamente il 20% degli utenti clicca sull'immagine. Per sfruttare questo tutte le immagini dovrebbero essere cliccabili con evento correlato, l'utente è abituato cliccarci sopra e si aspetta sempre qualcosa.

##### 5.2.4.1 Slideshow

Ascoltiamo ora un esempio pratico di problemi all'interno del sito già citato varie volte [www.jcrew.com](http://www.jcrew.com). Lezione tenuta dal professor Marchiori:

<https://drive.google.com/open?id=0B3-HN4UsFMNfRHZWZDVRaWhOU1E> (minuto 39:50).

#### 5.2.5 Lo spostamento dell'utente: Fitts

Se capiamo gli spostamenti dell'utente possiamo andargli incontro e incrementare la facilità d'uso del nostro layout con cui esso dovrà interagire. Fortunatamente una legge matematica viene in nostro soccorso.

##### 5.2.5.1 Legge di Fitts

Rappresenta il modello matematico del movimento umano in una singola dimensione (retta). La legge spiega quanto tempo impiega il cursore a spostarsi da un punto ad un altro.

$$T = a + b \cdot \log_2\left(1 + \frac{D}{W}\right) \quad (5.1)$$

dove:

- $T$ : è il tempo medio impiegato per concludere il movimento.
- $a$ : tempo costante impiegato dall'essere umano per cominciare e concludere l'azione.
- $b$ : covelocità, costante che dipende dagli strumenti utilizzati e dall'utente.

- $D$ : distanza dal punto iniziale alla zona obiettivo.
- $W$ : ampiezza della zona obiettivo.

Da notare che la distanza non conta molto nel tempo  $T$  finale, ma pesa molto di più l'ampiezza del target finale  $D$ .

**Past & click o drag & drop?** Vediamo subito un esempio pratico della legge di Fitts 5.1. Meglio *past & click* o *drag & drop*? Applicando la legge di Fitts abbiamo che la covelocità per il *drag & drop* è maggiore a causa del muscolo in tensione, per cui *past & click* è da preferire nelle interfacce web.

**Implicazioni della legge di Fitts** La principale implicazione che emerge è che gli **oggetti più grandi** sono **più facili** da raggiungere e cliccare. Ecco perché i menu sono odiati, non conta solo la distanza. I menu di navigazione devono essere più bilanciati. Altra implicazione è la *target size rule*: la taglia di un pulsante dovrebbe essere proporzionale alla sua frequenza d'uso. Un ottimo esempio dell'uso della *target size rule* è il redesign di Microsoft Office uscito nel 2007 (figura 5.3).



Figure 5.3: Il comportamento degli utenti - Microsoft Office dopo e prima il restyling

**Complicazioni** La legge di Fitts nasce considerando una dimensione, se l'area è uguale (variabile  $W$ ) ma ha forma diversa? Quale pulsante è il migliore? Il pulsante migliore dipende dall'angolo creato dal punto di partenza e la retta. Per cui attenzione!

#### 5.2.5.2 Interfacce e Fitts

Altri esempi più fini dell'applicazione della legge di Fitts possiamo trovarli già dal primo sistema Macintosh. Il menu è fissato distante, in alto in modo da non avere problemi con l'asse  $y$ . Risultato: 5 volte più veloci in media dei menu Windows da qui poi nasceranno le *task bar*. Ancora, lo strumento

di *scroll* su Mac ha una barra più lunga e i pulsanti sono vicini al contrario di Windows.

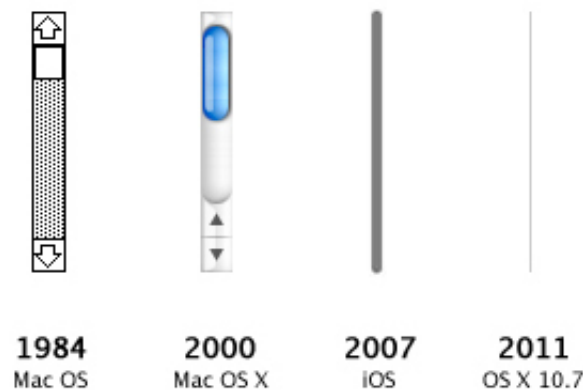


Figure 5.4: Il comportamento degli utenti - L'evoluzione della *scrollbar* su Macintosh

**Zone magiche** Altri suggerimenti che nascono dalla legge di Fitts riguardano i bordi e ancora meglio gli angoli. Essi possono essere utilizzati come pulsanti d'interfaccia super efficaci perché non necessitano del tempo di frenata e l'area dell'obiettivo è molto grande a patto che l'utente non abbia una finestra ridotta sullo schermo. La figura 5.4 mostra esplicitamente come il sistema operativo Os X abbia sfruttato queste zone magiche per la *scrollbar*.

**Menu 2.0** Anche nei menu possiamo vedere molti usi della legge di Fitts.

**Menu a pop-up:** è un'estremizzazione di Fitts, abbiamo i pulsanti vicinissimi al cursore del mouse.

**Menu circolare:** un'evoluzione del *menu a pop-up*, può essere completo (*pie menu*) o solo in parte, da usare negli angoli soprattutto (*fan menu*).

Tra i *pie menu* e i *menu lineari* non c'è dubbio che sull'usabilità vincono i primi, ma non sottovalutiamo i secondi che per tanti elementi/comandi se combinati assieme ad altri tipi di menu possono dare risultati efficaci. Ricordiamo che i primi software ad implementare i menu 2.0 sono stati i videogame, l'uso del *pie menu* in *Second Life* (figura 5.5 lo portò ad un gran successo. In altri l'uso dei *pie menu* permetteva di essere il più veloci possibili producendo eventi attraverso movimenti del mouse che aprivano catene di *pie menu*.

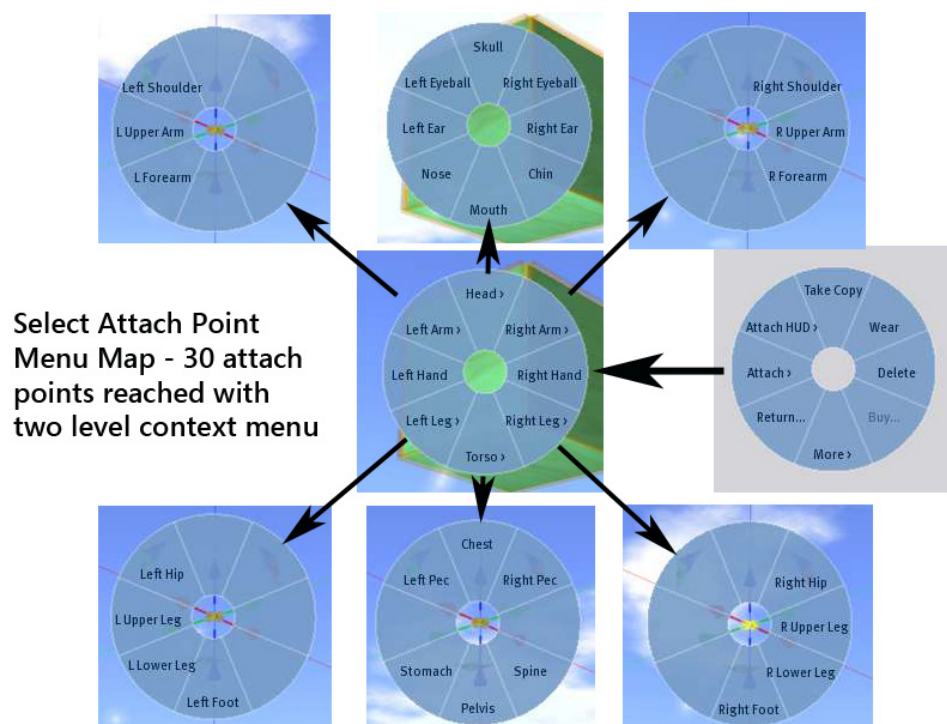


Figure 5.5: Il comportamento degli utenti - Pie menu su *Second Life*

## Chapter 6

# La pubblicità

Il modello di business classico è dare servizio gratis attraverso il web, ricevere utenti e sostenersi con la pubblicità. Piccolo intoppo: **gli utenti odiano la pubblicità**. Solo il 0,4% clicca sulle pubblicità. Cosa fare per migliorare questa situazione e soprattutto far sì che l'utente clicchi su essa?

- Buon posizionamento banner.
- Messaggio efficace, bello e attraente.

### 6.1 Posizionamento

Per quanto riguarda il posizionamento del o dei banner in coordinate assolute, per ordine di utilità, abbiamo:

1. Colonna di sinistra.
2. Top della pagina.
3. Colonna di destra.
4. Bottom (non serve praticamente a nulla).

Altri accorgimenti da adottare possono essere:

- Posizionare la pubblicità vicino al contenuto interessante così, maggiori visualizzazioni per essa.
- Rispettare le taglie minime e tener conto che la taglia non influenza molto. I banner grandi non attraggono molto di più di quelli piccoli.

### 6.2 Messaggio efficace

Prima di analizzare come fare un messaggio pubblicitario efficace mostriamo le **11 cose da non fare** con percentuale di insoddisfazione generata agli utenti.

### 6.2.1 Top 11 disgrazie

11. Suona automaticamente (79%).
10. È in movimento (79%).
9. Lampeggia (87%).
8. Occupa la maggior parte della pagina (90%).
7. Si sposta sullo schermo (92%).
6. Non dice di cosa si tratta (92%).
5. Copre il contenuto da leggere (93%).
4. Non dispone di un modo chiaro per chiudersi (93%).
3. È qualcosa che cerca di farti cliccare sopra (94%).
2. Si carica lentamente (94%).
1. È un pop-up (95%).

### 6.2.2 Il mondo pubblicitario nel web

Per risaltare il messaggio nel mondo pubblicitario esistono accorgimenti molto efficaci come, ad esempio:

- Utilizzare persone belle per attirare l'attenzione.
- Usare colori vivaci ed effetti speciali.

Questi accorgimenti trasportanti nel web rimarranno efficaci?

#### 6.2.2.1 Effetto zapping

Ricordiamo che le immagini nel web godono di meno visibilità e attenzione da parte dell'utente rispetto al testo, questo deriva da un processo automatico e subconscio (effetto *zapping* nel web) che salta le immagini e soprattutto i banner. Portare infatti un contenuto in forma simile a immagini e banner può far sì che questo non sia nemmeno visualizzato dalla maggior parte degli utenti.

Gli utenti sono abituati a ritenere superflue le informazioni contenute nei banner per cui durante la navigazione il cervello utilizza dei filtri (che utilizziamo quotidianamente anche in altre situazioni) sull'informazione entrante dall'apparato visivo.

### 6.2.3 Immagini in serie A

L'unico modo, paradossale, per rendere attrattive le nostre immagini e quindi banner è **andare controcorrente**. L'algoritmo abituale per filtrare è interrotto da qualcosa di inusuale e l'utente è costretto a porre maggiore attenzione. Quindi:

- Niente colori vistosi e attraenti.
- Confondere le idee all'utente.

Per far ciò si può ricorrere ad alcune tecniche:

**Blending:** eliminare le zone della pubblicità confondendo pubblicità con contenuto. Il miglior *blending* è il testo.

**Giochetti web:** l'uso di giochi attira l'attenzione e azzeri i timer dell'utente finché è impegnato a giocare. Una buona tecnica quindi è quella di inserire pubblicità insieme a giochi.

Attenzione però al *distraction effect*: la pubblicità deve seguire il contesto in cui è contenuto altrimenti i timer dell'utente diminuiscono del 40% e la voglia di rivisitare il sito scende addirittura dell'80%.

### 6.2.4 Il problema del contenuto

Per risolvere il problema del contenuto e del *distraction effect* si è ricorso al *behavior advertising*, pubblicità che seguono il comportamento dell'utente nel web e che quindi è più probabile siano interessanti. Per far questo è necessario raccogliere dati su ogni singolo utente. Questo può rendere fino a 10000 volte più efficace la pubblicità e in certi casi ribalta l'insoddisfazione con aumento dei timer e voglia di ritornare al sito.



## Chapter 7

# La ricerca

Un altro problema fondamentale dei siti è far sì che il proprio contenuto sia trovato dagli utenti. La ricerca diviene quindi un fattore fondamentale all'interno del web. Esistono varie modalità di ricerca, prima di tutto ogni sito internet se abbastanza grande dovrebbe avere come servizio la ricerca interna vedremo come inserirla nel nostro sito garantendo l'usabilità ed discutendo problemi di molti siti. Scopriremo anche i motivi per cui diverse modalità di ricerca al primo impatto stupefacenti si rivelano essere fallimentari per l'usabilità dei siti internet. Sono le ricerche 2.0 che con numerosi tentativi negli anni hanno cercato di sfruttare la sintesi vocale lottando contro il problema del contesto.

### 7.1 Ricerca interna

Una ricerca interna al proprio sito e la qualità di essa è determinata dal numero di pagine che questo ha.

- **>100** è **necessario** che il sito abbia uno strumento di ricerca.
- **>1000** è **cruciale** che lo strumento di ricerca sia eccellente e copra tutto il sito.

Si stima che la ricerca è utilizzata quasi nel 100% dei casi. Gli utenti sono abituati ai motori di ricerca e molto spesso gli utenti arrivano in una pagina qualunque del sito attraverso il *deep linking*. Difatti:

- il 60% usa la ricerca appena giunge in un sito (se essa è disponibile).
- il 40% fa uso di link.

Vista l'enorme importanza in molti siti si tende ad inserire uno strumento di ricerca, ma questo se fatto come si deve è molto costoso e troppo spesso si incorre all'uso di soluzioni *low cost* che peggiorano soltanto la situazione. Solitamente questa soluzione è di localizzare motori di ricerca sul proprio sito: costo zero ma:

- l'algoritmo dei motori di ricerca è ottimizzato per larghe scale, su bassa scala non dà risultati soddisfacenti.
- Se l'utente usa la ricerca significa molto probabilmente che il motore di ricerca già utilizzato non l'ha portato al suo scopo per cui utilizzarlo ancora non servirà a niente.
- I motori di ricerca non fanno la lettura di tutte le pagine (il 70% del web è sconosciuto ai motori di ricerca) per cui questo può far sì che alcune pagine del sito non compariranno mai tra i risultati.

### 7.1.1 Ricerca su misura

Gli utenti preferiscono:

- 1 sola modalità di ricerca come i motori di ricerca.
- Un **box testuale** e un **pulsante** con scritto "search" (o "cerca"). Null'altro è da sostituire alla parola "search".
- Si può sostituire il box testuale e il pulsante con l'icona della lente per siti mobile. In generale però gli utenti preferiscono sempre il testo.

#### 7.1.1.1 Principio di Robert Browning

Enunciamo un principio che nel web ha praticamente valenza generale. Il primo ad annunciarlo fu Robert Browning e in seguito fu ripreso dall'architetto modernista Ludwig Mies van der Rohe.

*“Less is more”*

Che applicato al web consiste in: *meno dettagli e orpelli significa qualcosa di meglio*. L'esempio lampante è la pagina iniziale del motore di ricerca Google. Ludwig Mies van der Rohe estese anche il principio in:

*“Less is more but God is in the detail”*

Per capire quanto i dettagli sono importanti si veda l'Errore 404 spiegato successivamente.

## 7.2 Ricerca vincolata

La ricerca vincolata si prefigge di accompagnare l'utente passo passo attraverso parametri, deve essere considerata come un'aggiunta alla classica ricerca spiegata in precedenza. Vantaggi e svantaggi:

- è efficiente e gradita dagli utenti.
- esistono due modalità di come si offre ognuna con i suoi svantaggi.

### 7.2.1 Ricerca vincolata dinamica

La ricerca dinamica avviene per ogni settaggio di un parametro. Svantaggi:

- Quando il numero dei vincoli è tanto l'attesa del caricamento è alto.
- Abbiamo un carico del server notevolmente maggiore.
- Se l'utente ha già i vincoli in mente dovrà fare comunque più ricerche.

### 7.2.2 Ricerca vincolata statica

La ricerca avviene solo nel momento in cui l'utente ha impostato tutti i parametri richiesti e ha dato il via. Svantaggi:

- Il pulsante non ha contenuto standard. Le parole "search" o "cerca" illudono che sia la ricerca classica.
- Se un utente è abituato a quella dinamica resta confuso al prima impostazione di parametro perché non succede nulla. Questo può indurlo a pensare che non funzioni.

### 7.2.3 Soluzione ibrida

In generale la soluzione dinamica è meglio nel caso in cui un utente non sappia cosa cercare mentre p il contrario per la ricerca statica. Esiste una terza soluzione che consiste nell'avviare la ricerca automaticamente al termine del settaggio dei parametri. In questo modo togliamo il pulsante per la ricerca statica e non appesantiamo il server con ricerche in più.

## 7.3 Consigli

DI seguito elenchiamo alcuni consigli per mostrare i risultati di ricerca nel migliore dei modi.

**Ordinamento:** mettere sempre a disposizione un ordinamento bidirezionale.

**Casi limite:** se non ci sono risultati dobbiamo comunicare all'utente che ci non ci sono risultati altrimenti si confonde l'utente che pensa non funzioni.

**Error 404:** quando i link sono "rotti" (non esistono più) deve esserci una pagina con suggerimenti all'utente che offrono aiuto. Molte volte a queste si affianca qualcosa di simpatico che ironizza la situazione, ciò sospende i timer, un esempio di questo lo troviamo nella pagina 404 del sito [www.b3ta.com](http://www.b3ta.com), che raccoglie le più esilaranti immagini delle pagine error 404 nel web.

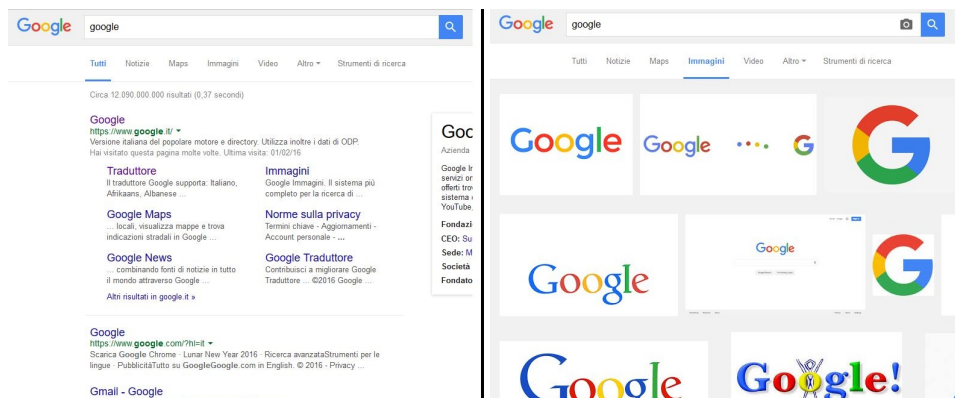


Figure 7.1: La ricerca - Presentazione dei risultati lineare e a griglia

**Presentazione risultati:** esistono due modi per presentare i risultati di una ricerca.

- Modalità classica: vedi figura 7.1.
- Modalità a griglia: permette di visualizzare molta più informazione compatta ma questo fa sì che gli elementi mostrati perdano rilevanza. Con questa modalità l'utente si confonde e si perde nella micro-navigazione. Non a caso Google immagini propone i risultati a griglia, vuole che l'utente stia per più tempo all'interno del motore di ricerca. (vedi figura 7.1)

## 7.4 Il search box

Agli albori del web nelle caselle di ricerca si inserivano nella maggior parte dei casi parole uniche, con il passare del tempo però gli utenti usano sempre più parole. Il numero massimo di caratteri determina il numero di utenti a cui basterà.

- con 10 caratteri solo il 35% degli utenti sarà soddisfatto.
- con 30 l'88%.

Quest'ultimo numero è la media consigliata. Un box troppo piccolo aumenta lo stress proporzionalmente per ogni carattere che sfiora e gli utenti tendono a scrivere meno con risultati della query più scarsi. Per la necessità di spazio per il design in molti casi si è ricorso ad un **box dinamico** che una volta cliccato si allarga per contenere più caratteri.

## 7.5 Ricerche 2.0 (\*)

(N.B. la seguente sezione è incompleta)

Il web è un media informativo visivo. Negli anni sono stati vari i tentativi di evolvere dal testo alla voce per un'interazione uomo-macchina di livello superiore. Esistono varie tecnologie:

- VoiceXML
- SSML
- PLS
- CCXML
- SRGS

che via via hanno migliorato l'interazione uomo macchina ma si è ancora lontani da *HALL 9000* il supercomputer intelligente di *2001: Odissea nello spazio*.

### 7.5.1 Il grande problema

Basarsi sulla voce non basta perché rende difficile interpretare le parole senza l'aiuto del **contesto**. La tecnologia *Google Glass* non ha avuto successo proprio per questo motivo, l'utenza chiedeva più pulsanti perché la sintesi vocale non era ancora pronta.

### 7.5.2 L'utente e le ricerche 2.0

Le ricerche del 2.0 diminuiscono la soddisfazione del 42%! Perché?

- Di fronte ad un assistente virtuale l'utente si forma aspettative più alte e per questo diventa ancora più esigente.
- Illudere l'utente presentando un robot simile all'umano non aiuta, molto meglio mantenere un'interfaccia robotica per non dare illusioni e creare grosse aspettative.

Un esempio positivo che segue questi accorgimenti è *Anna* di IKEA. Vediamone i punti forti:

- riceve in input solo testo e presenta un engine linguistico ben fatto.
- Ha un'interfaccia cartoon.
- È uno strumento facoltativo.

### 7.5.3 Ivoluzione/evoluzione dell'interazione

Per il problema del contesto nei videogiochi si è sempre più limitata la libertà di scelta e interazione negli anni. Ciò perché un grado di libertà alto poteva generare incomprensioni tra utente e macchina e la frustrazione aumentava.

#### 7.5.3.1 Il rumore

Un altro motivo per cui le interfacce non precise hanno un gradimento superiore è il rumore di fondo. È grazie al rumore di fondo che sembrano più naturali. A tal proposito si sono fatti studi sul rumore che costantemente viviamo. Esso ha una natura frattale ricorsiva e per generarlo basta continuare a **ridurre l'ampiezza** e **alzare la frequenza** e infine mettere insieme tutte le parti. Paradossalmente le interfacce imperfette danno più soddisfazione.

## Chapter 8

# La visibilità

Altra caratteristica essenziale per l'informazione all'interno di un sito è l'essere trovata. Per far ciò conta la **visibilità esterna** e questa è determinata dal posizionamento del sito sui motori di ricerca. Vedremo che è essenziale posizionarsi nella top 10 per le *keyword* che ci riguardano e capiremo come promuovere un sito analizzando il funzionamento dei *search engine* (SE) e l'uso che gli utenti ne fanno.

### 8.1 Posizionamento SERP

La top 10 risultati assorbe il 95% dei click dell'utenza dopo una ricerca.

1. 51% click (il primo risultato prende più di tutte le altre!),
2. 16% click;
3. 6% click;
4. 6% click;
5. 5% click;
6. 4% click;
7. 2% click;
8. 1% click;
9. 1% click;
10. 2% click (effetto *black jersay* o *Malabrocca*).

Nota: le strategie che verranno illustrate in seguito non funzionano sempre, basta infatti un minimo cambiamento per cambiare il comportamento degli utenti sui risultati. Infatti recenti introduzioni di Google sono state

analizzate molto, prima di farle diventare effettive. La striscia di immagini per esempio svolgono un effetto sospensivo per trattenere l'utente e non hanno impatto sulla percezione del testo.

## 8.2 SPAMDEX (SEO)

È l'insieme di attività svolte con lo scopo di far sì che un sito sia indicizzato da Google e occupi le posizioni più alte della SERP. Nelle seguenti sezioni prima spiegheremo il funzionamento dei motori di ricerca e poi illustreremo le tecniche per migliorare il posizionamento o addirittura ingannare il SE. Il **punteggio di una pagina** è calcolato sul riferimento a due componenti delle pagine:

- componente **testuale**;
- componente **ipertestuale**.

### 8.2.1 Parte testuale

Per la parte testuale si misura la qualità in base a quanto una parola è importante per una pagina attraverso il TFIDF. Per misurare l'importanza di una parola si utilizzano due tecniche:

#### 8.2.1.1 Term frequency (TF)

Misurare la frequenza di una parola. Ad esempio su 100 parole in una pagina 5 sono di nostro interesse per cui il TF è uguale al 5%. È ovvio che solo questa tecnica non basta per avere risultati soddisfacenti. Parole molto frequenti come articoli, congiunzioni, eccetera avrebbero un'importanza troppo alta anche se prive di significato.

#### 8.2.1.2 Inverse document frequency

È la misura dell'inverso della frequenza della parola nell'insieme dei documenti (il sito web) scalata logaritmicamente.

Esempio: su 1000 pagine "il" appare in 980 pagine, ha TF=98% mentre:

$$IDF = \log \left( \frac{1}{0,98} \right) = 0,008;$$

Altro esempio: su 1000 pagine la parola "pippo" ha il 10% di frequenza, allora:

$$IDF = \log \left( \frac{1}{0,1} \right) = 1;$$

Ora più rara è la parola più è alto il bonus.



### 8.2.1.3 Query SE

Ogni *query* sarà composta dalle parole con cui il SE andrà a calcolare TF e IDF nei siti web. Quelli con il valore più alto si posizioneranno più in alto. Se sottoponiamo una *query* con una parola  $p$  il SE prende le pagine dove appare  $p$  e ne calcola il TF e l'IDF. Per più parole in una *query* si usa la somma del TF e IDF. In questo modo alzare il punteggio di una parola può rischiare di diluire il valore e viceversa. È importante quindi selezionare un gruppo di parole e alzare il loro TFIDF a scapito di altre, queste *keyword* saranno le parole che competeranno nella SERP.

### 8.2.2 Term spam

Vediamo quindi alcune tecniche per aumentare il punteggio per le *keyword* definite.

**Body spam:** inserire le *keyword* semplicemente nel body html.

**Title spam:** inserire le *keyword* sul titolo delle pagine web, è una parte che l'utenza media non vedrà mai.

**Meta tag spam:** inserire le *keyword* in appositi tag html (i *meta*) che descrivono a *keyword* la pagina. Per l'inteso sfruttamento i SE oramai ignorano questo genere di spam.

**Anchor text spam:** inserire le *keyword* nel testo di un link tra i tag `<a>...</a>`. Questo oltre al punteggio della pagina aumenta il punteggio alla pagina destinazione e il bonus viene dato con meno limiti per il TFIDF.

**URL spam:** inserire le *keyword* nell'URL dell'indirizzo della pagina. Otteniamo bonus simili a quelli dell'*anchor text spam*.

#### 8.2.2.1 Starter kit

Abbiamo visto dove inserire le *keyword*. Ora vediamo le tecniche utilizzate di come inserirle:

**Repetition:** ripetere la *keyword* stando attenti al suo TFIDF. Tecnica facile da trovare dal SE che può pure penalizzare.

**Dumping:** inserire termini rari nel testo anche se non c'entrano nulla. Otterremo un punteggio alto sul sito anche se questo non è pertinente.

**Wearing:** utilizzare pezzi di altre pagine web in cui sono inserite le nostre *keyword* in modo random.

**Stitching:** *past & copy* da siti differenti e si assembla il tutto ottenendo un contenuto rilevante. Per esempio da una risposta di un utente su un forum sviluppiamo un articolo nuovo. Più argomenti diversi danno inoltre un bonus globale e i SE fanno fatica a capire se è contenuto copiato.

**Broeding:** oltre alle *keyword* fare uso di sinonimi e frasi correlate. Questo coprirà più le *query* degli utenti.

#### 8.2.2.2 Quali keyword?

Una domanda di cui la risposta probabilmente decreterà il successo di un sito. Le *keyword* devono essere pensate in base a quello che cerca l'utente. Occorre un'attenta analisi prima di lanciarsi nel mondo SEO. Di seguito alcuni strumenti utili a tale scopo:

- [Google keyword tool](#) (a pagamento);
- [keywordtool.io](#) (alternativa gratis);
- [Google Trends](#) (ottimo per capire la tendenza nel tempo);

### 8.2.3 Problema e soluzioni del term spamming

Non basta potenziare la pagina, se infatti questa non è utile e soffre di problemi di usabilità l'utente semplicemente abbandonerà. Se vogliamo potenziare una pagina con le tecniche di *term spamming* dobbiamo nascondere in qualche modo all'utente ciò che serve ai SE.

#### 8.2.3.1 Content hiding

Il contenuto potenziante è nascosto all'utente e solo i motori di ricerca lo vedono.

#### 8.2.3.2 Redirection

Anche detta tecnica 302, consiste nel potenziare una pagina che sarà costruita con soli fini di posizionamento che indirizza alla vera pagina. In questo modo gli utenti vedranno la pagina studiata per loro mentre i SE si fermeranno a leggere la prima. Questa tecnica funziona solo con trucchetti che richiedono javascript, il *meta* che reindirizza infatti è riconosciuto dai SE. Con l'uso di javascript il *meta* verrà aggiunto dinamicamente mentre i SE, per costi di risorse, non eseguono il codice. Tuttavia ad oggi i SE leggono il codice javascript e cercano se esiste questa tecnica, per ingannarli è sufficiente offuscare un po' il codice.

### 8.2.3.3 Cloaking

Controlla l'accesso alla pagina per capire se chi visita è un bot o un umano. Se questo è un bot li presento una pagina apposita diversa da quella mostrata all'utente. È una tecnica difficile da scoprire tanto che Google la penalizza moltissimo se segnalata ma questo avviene solo tramite intervento umano.

## 8.2.4 Parte ipertestuale: pagerank

Il punteggio delle pagine web è affidato anche ai link di ipertesto: il *pagerank*. Ad ogni pagina  $\pi_v$  era affidato un punteggio pari a:

$$\pi_v = \sum_{(\omega, v) \in E} \frac{\pi_\omega}{d_\omega}$$

Dove:

- $\omega, v$  è un link da  $\omega$  a  $v$ ;
- $d_\omega$  è il numero di link uscenti.

Ossia, se una pagina  $\omega$  ha un link verso un'altra pagina viene data una frazione ( $1/d_\omega$ ) di *pagerank* a quest'ultima. Come vincolo abbiamo:

$$\sum \pi_v = 1$$

Questo calcolo di *pagerank* molto naive era molto facile da manipolare con gli stratagemmi visti in precedenza, si è deciso quindi di farne una riformulazione.

### 8.2.4.1 Le catene di Markov e il random walk

La riformulazione avviene sfruttando l'algebra lineare e speciali matrici dette catene di Markov che godono di alcune proprietà unite alla teoria della probabilità. Il *pagerank* di una pagina web diventa quindi la probabilità di entrare in quella pagina da altre pagine. Per rendere più chiaro il concetto si può pensare ad una scimmia (un *random surfer*) che naviga il web in modo del tutto casuale, la probabilità che la scimmia entri in una pagina ne determina il suo punteggio.

Esempio: dato il seguente grafo i nodi che rappresentano pagine e gli archi i link:  $A \longleftrightarrow B \longleftrightarrow C$ ,

B ottiene il 50% del flusso mentre A e C il 25%.

Il *random surfer* è un altro fattore su cui tener conto ma attenzione esso non rispecchia esattamente gli intenti degli utenti che non navigano totalmente a caso.

#### 8.2.4.2 Problemi: spider traps e island

Per attuare il sistema spiegato in precedenza c'è la necessità di far girovagare questi *random surfer* (i famosi *spider*) che raccolgono dati da cui calcolare il *pagerank*. La navigazione casuale da pagina a pagina può rivelarsi fatale in alcuni casi.

**Spider traps:** nel caso in cui uno spider rimanga intrappolato all'interno di un sito in cui pagine continuano a indirizzare ad altre pagine in un ciclo infinito (si pensi ad un calendario online).

**Island:** esistono poi casi in cui alcuni pezzi di rete sono isolati dagli altri, dei veri e propri micro-web indipendenti in cui per gli *spider* non c'è alcun modo di passare, ciò significa: nessun valore di *pagerank* o infiniti valori diversi di *pagerank*.

#### 8.2.5 Pagerank 2.0

A seguito dei problemi prima elencati si è cercato un'altra soluzione: il teletrasporto. La formula quindi cambia in (con  $N$  numero totale di pagine):

$$\pi_v = (1 - \epsilon) \cdot \left( \sum_{\omega, v \in E} \frac{\pi_\omega}{d_w} \right) + \frac{\epsilon}{N}$$

Nella pratica si è aumentata la casualità. Ora la scimmia ha anche un dado, con una certa probabilità effettua il teletrasporto in un'altra pagina casuale altrimenti fa come spiegato in precedenza (da notare un comportamento apparentemente più strano se paragonato con quello degli utenti, in realtà non poi così tanto poiché molte volte un utente effettua il teletrasporto attraverso URL diretto).

In questo modo grazie al **fattore di teleporting**  $\epsilon$  risolviamo *spider traps* e *island*; più questo fattore è verso il 100% e più il punteggio in probabilità alle pagine diventa uguale per tutte, viceversa più è verso lo 0 e più il *pagerank* si avvicina a quello calcolato in precedenza.

Un altro modo per il calcolo del *pagerank* è il *total rank*:

$$T = \int_0^1 r(\alpha) d\alpha$$

##### 8.2.5.1 Problema: dead ends

Nonostante l'introduzione del teletrasporto rimane il problema (prima non illustrato) delle *dead ends*. Con questa formula l'egoismo paga, un sito è meglio che mantenga i link al suo interno.

La soluzione a cui sono arrivati i moderni motori di ricerca è calcolare l'importanza su altro per ogni sito e pagina web e non sul vero web. (vedere sezione Web semantico)

## 8.2.6 Migliorare il pagerank

Esistono due fattori fondamentali che influenzano il *pagerank* con l'uso del teletrasporto:

- Inlink;
- Outlink.

### 8.2.6.1 Inlink

Link entranti alla pagina. Per incrementare il *pagerank* di una pagina dobbiamo fare in modo che questa sia puntata dal maggior numero di pagine esterne. Alcune tecniche per fare ciò sono:

**Infiltration:** infiltrarsi in altri siti e mettere link alla pagina da potenziare (commenti spam etc.).

**Honey pot:** creare contenuto appetibile e utile che riceve link (è il modo giusto di aumentare il *pagerank*). Molto spesso questo contenuto è vittima del *copy & past*.

**Link exchange:** allearsi con altri per scambiarsi link.

**Resurrection:** acquistare domini defunti con *pagerank* abbastanza alto. Il *pagerank* infatti non decade mai e resta legato al dominio, inoltre l'età del dominio conta come bonus. Esistono interi business per la vendita di domini defunti ma appetibili per il proprio *pagerank*.

**Marketing 2.0** Un esempio nella storia del web di quanto siano importanti gli *inlink* arriva dalla pagina [million dollar page](#). Una schermata di pixel venduta a 1 dollaro l'uno, il punto forte per cui tutti si affrettarono di acquistare almeno 1 pixel fu la presenza di tutti i link che puntavano alla pagina resa famosa dall'originale idea dell'autore. La pagina diventò una vera e propria cascata di flusso per il *pagerank* ed ogni link al suo interno ne acquistava un po' ad un prezzo ridicolo.

### 8.2.6.2 Outlink

Passiamo ora ai link uscenti. Posizionando un link esterno in una pagina un po' di flusso di *pagerank* esce dalla pagina. In questo modo è garantita la **solidità**, l'aggiunta di *outlink* non alza il punteggio e lo *smadex* non può essere locale. Questo controintuitivamente non risulta essere vero! Con la presenza di *outlink* posso aumentare il *pagerank* della mia pagina con dinamiche imprevedibili. Ciò deriva dalla presenza del *fattore di teleporting*.

### 8.2.6.3 Spam farm

Esistono numerose tecniche per usare gli *outlink* e gli *inlink* per aumentare il *pagerank*. Le *spam farm* sono una di queste tecniche: strutture apposite studiate di link e pagine per l'incremento del punteggio nei vari SE. Una *spam farm* ottimale singola consiste in una pagina o più pagine, definite pagine **target**, puntate da link bidirezionali da altre pagine, definite pagine **potenzianti**. Un struttura ottimale inoltre presenta la proprietà di *reachability*, ovvero se gli spider dei SE raggiungono la pagina *target* attraverso una pagina *potenziante* allora tutta la *spam farm* diventa raggiungibile (vedere figura 8.1).

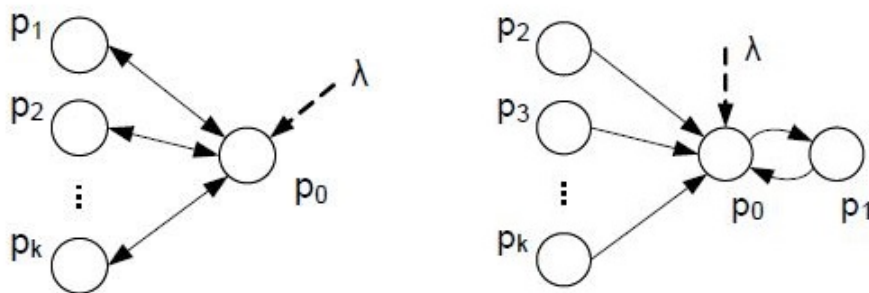


Figure 8.1: La visibilità - Esempi spam farm singole ottimali

**Alleanze** Il modo migliore per sfruttare le *spam farm* è sfruttare alleanze con altri siti internet. Le alleanze tra siti (due e più) possono essere di tipo:

**Alleanza profonda** Le pagine target distribuiscono il flusso sulle pagine potenzianti dell'altro sito (figura 8.2) creando una media dei *pagerank* stabile.

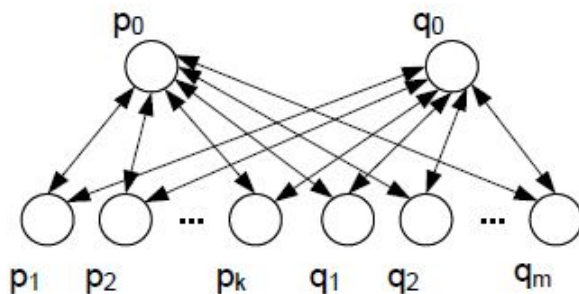


Figure 8.2: La visibilità - Spam farm alleanza profonda

**Alleanza superficiale** Le pagine target dei due siti sono connesse con link (figura 8.3, il *pagerank* ottenuto risulta essere più del massimo delle due pagine target prima della connessione.

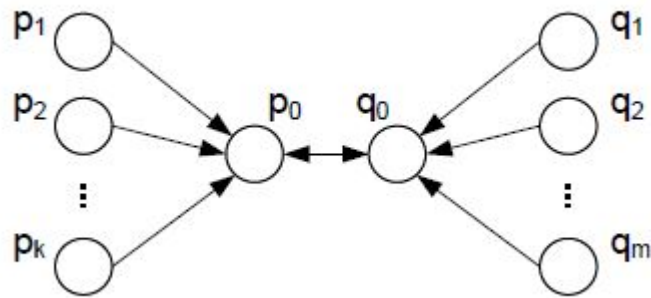


Figure 8.3: La visibilità - Spam farm alleanza superficiale

**Alleanze Ring** Alleanza che comprende più di due siti. Le pagine target indicizzano in un percorso circolare la pagina target successiva creando una sorta di anello (figura 8.4).

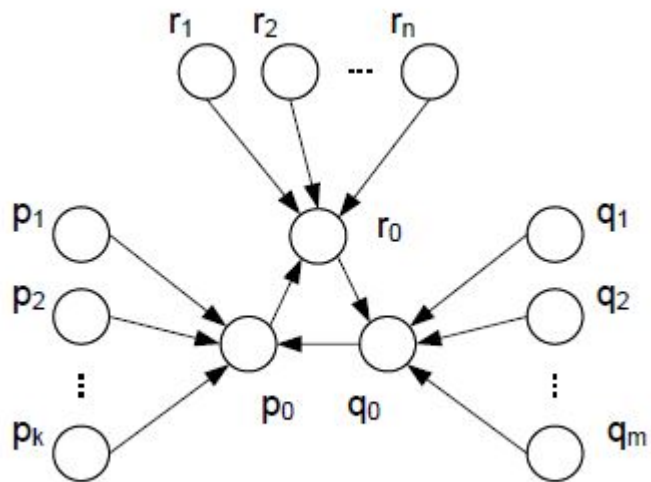


Figure 8.4: La visibilità - Spam farm alleanza ring

**Alleanza complete core** Potenziamento della *spam farm* di tipo *ring*, la struttura bidirezionale è più vantaggiosa. Ora ogni pagina target ha link

entrante e uscenti verso le due pagine target più vicine (figura 8.5). Per fare un core basta un grafo fortemente connesso.

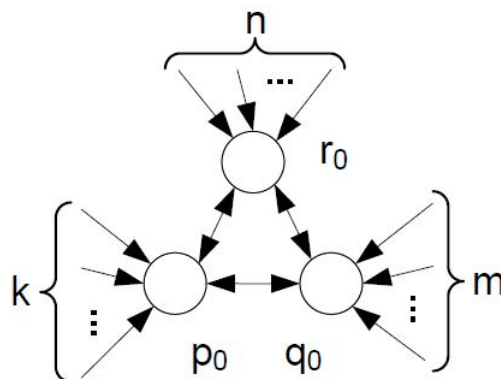


Figure 8.5: La visibilità - Spam farm alleanza complete core

#### 8.2.6.4 Contromisure SE

Ovviamente le *spam farm* sono azioni penalizzate dai SE poiché ingannano la misura del *pagerank*. I motori di ricerca controbattono quindi cercando di identificare queste unioni rivelando l'esistenza di grafi fortemente connessi. Questi grafi dipendono dal modo con cui l'alleanza si è creata e dal numero di partecipanti ad asse. Visto il numero enorme di combinazioni fu creata un'enciclopedia delle sequenze. Purtroppo in ogni sequenza il numero di collegamenti da controllare cresce enormemente in base al numero di partecipanti ( $N$ ), per:

- $N=3 \rightarrow 18$  collegamenti;
- $N=4 \rightarrow 1606$  collegamenti;
- $N=5 \rightarrow 565080$  collegamenti;
- ...

Non esiste nessuna formula che calcoli il risultato di collegamenti dato un numero  $N$  di partecipanti. Inoltre il costo computazionale risulta dispendiosissimo. Si sono quindi cercate soluzioni "lateralali" più sofisticate.

Un modo consiste in calcolare il vecchio *pagerank* e confrontarlo con quello nuovo. Se il rapporto tra le due misure risulta troppo alto significa che la pagina analizzata è potenziata. È stato stimato che il successo di questa tecnica si trova tra il 95% e 100%.



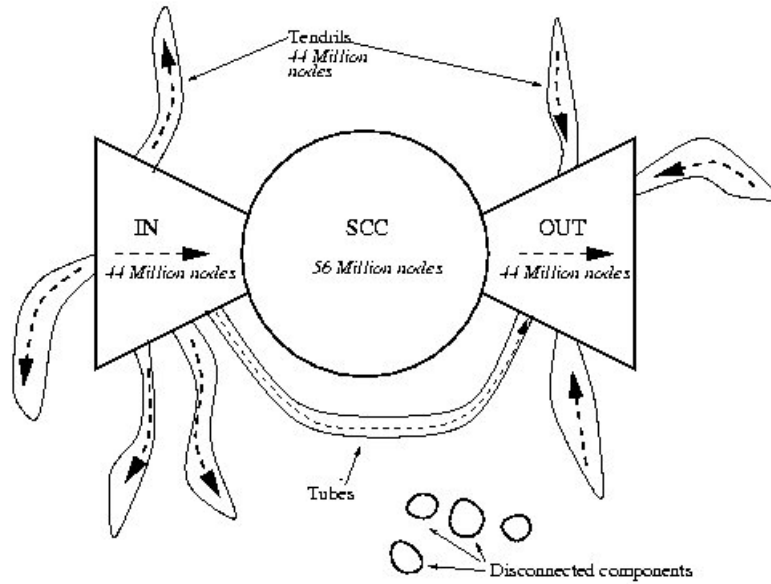
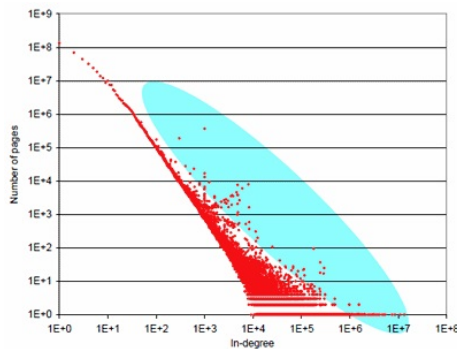
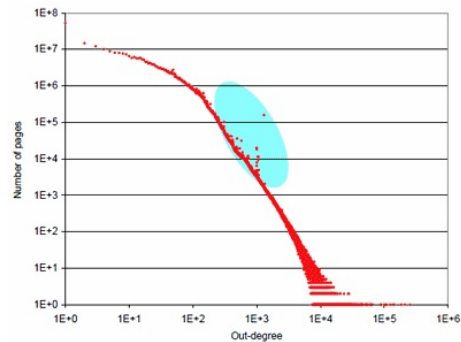


Figure 8.6: La visibilità - Forma ad alto livello del web

Un altro modo consiste nel basarsi sulla **forma del web**: la struttura di alto livello a papillon (figura 8.6). L'idea è di analizzare la forma di un sito web, se questa non rispecchia ed ha valori al di fuori della media nella norma globale dei siti web allora il sito è sospetto e avviene un controllo più approfondito. Nella figura 8.7a vediamo la zona creata dai link entranti nei siti nel web globale che rappresenta la norma. I punti situati all'esterno (evidenziati dall'ellisse) sono siti di cui si sospetta il potenziamento. Nella figura 8.7b invece vediamo i link uscenti.



(a) Distribuzione link entranti.



(b) Distribuzione link uscenti.

Figure 8.7: La visibilità - Rappresentazioni grafiche dei link nel web

### 8.2.7 Pagerank 3.0

Negli ultimi anni si è voluto migliorare la formula del *pagerank* in particolare il fattore di teleporting.

La vecchia formula (qui riporta in una versione leggermente diversa):

$$(1 - \epsilon) \cdot 1^T N + \epsilon L$$

possiamo scriverla usando la matrice  $P$  e diventa:

$$(1 - \epsilon) \cdot 1^T N + PL$$

$P$  è una matrice che permette il *pagerank* personalizzato che sostituisce il teletrasporto. Ogni nodo è pesato con preferenze così rendiamo il teletrasporto meno stupido. In questo modo però sorgono problemi riguardanti la libertà e la censura. Se infatti l'utente delle preferenze è il SE di turno, potrebbe volutamente alzare il *pagerank* ad alcuni siti e abbassarlo ad altri.

È noto che Google personalizza la bontà di un sito, alcune sono addirittura fatte a mano! *Searchking*, competitor di Google, è stato penalizzato in questo modo e il ricorso legale ha rivelato come il *pagerank* sia soltanto un'opinione. Con questo nuovo livello di *pagerank* quindi possiamo dichiarare defunta la democrazia contenuta in esso.

#### 8.2.7.1 Pagerank personalizzato

Il *pagerank* personalizzato però non è disponibile soltanto ai SE ma anche all'utente. Ogni profilo personale di ogni utente calcola un *pagerank* personalizzato in base a tutti i dati raccolti su di lui (vedi tutte le piattaforme Google disponibili che ogni giorno utilizziamo tutti). In questo modo i risultati di ricerca varieranno in base ai suoi interessi. Ma come calcolare questo *pagerank* personalizzato per ogni utente nel web!? Se restringiamo a soli due valori: "sì" e "no" abbiamo con  $N$  pagine:  $2^N$  personalizzazioni. Troppe... Il *pagerank* per fortuna ha la proprietà di essere **linearmente componibile** rispetto le probabilità; cioè se ho 100 pagine calcolate e risultanti "sì" e da un'altra parte ho altre 100 pagine calcolate e risultanti "sì" per calcolare il *pagerank* delle 200 pagine mi basta unire i risultati. Grazie a questa proprietà possiamo calcolare per  $N$  pagine  $N$  *pagerank* personalizzati. Sono ancora troppi calcoli...  $N$  è la taglia del web, miliardi di miliardi di pagine web!

La soluzione quindi è il *topic pagerank*, selezionare un certo numero di *keyword* e *topic*. Creare dei profili che approssimano gli utenti e su questo numero limitato calcolare il *pagerank* personalizzato diviso a categorie a sua volta divise in *topic* più precisi secondo le necessità. Combinando i *pagerank* precalcolati traccio il profilo degli utenti. Da questi profili "grezzi" si calcolano i profili astratti che li conterranno e andranno a modificare il *pagerank* in base alle ricerche dell'utente. Si fa notare che ogni *pagerank* personalizzato è compatibile con le contromisure ma comporta disturbi e limitazioni alle ottimizzazioni SEO.

### 8.2.7.2 Bontà di una pagina: doppio ranking

Fino a poco tempo fa il *pagerank* costituiva un punteggio positivo che co-esisteva con le contromisure dei SE per penalizzarlo. Le alterazioni e gli attacchi nel web però resero le contromisure non più sufficienti per questo si è deciso di affiancare al normale *pagerank* positivo un *pagerank* opposto: un punteggio negativo.

### 8.2.7.3 Janus Graph

Per ricorrere a questa nuova misura bisognava mantenere il modello sottostante sempre uniforme e semplice, si è quindi ricorso al **grafo di Giano**. Il *Janus Graph* è un grafo in cui ogni nodo ha due valori associati:

- $W^+$  per la parte positiva;
- $W^-$  per la parte negativa.

$W^+$  e  $W^-$  sono due dati totalmente scorrelati. Il *pagerank* finale è ottenuto dalla combinazione della parte buona e dalla parte cattiva, rispettivamente:

$$R_{good}(G^+) \ \& \ R_{bad}(G^-)$$

La combinazione è lineare, per cui:

$$\alpha R_{good}(G^+) - \beta R_{bad}(G^-)$$

Con  $\alpha$  e  $\beta$  pesi costanti.

Come calcolare però le due funzioni di *ranking*? Quali valori dare ai due pesi individuati?

### 8.2.7.4 Parte buona & parte cattiva

Soluzione al problema di **doppio ranking**: Dato un *ranking*  $R$  definiamo *janus extension*  $R^J$ . Le due parti quindi sono **simmetriche** e **duali**, entrambe riflettono la struttura del web. Invertiamo la struttura del web e identifichiamola con  $G^\#$ , se un link andava da A a B adesso va da B ad A. Calcoliamo  $R^J$ :

$$R^J = R(G^+) - R((G^\#)^-)$$

Dove:

- $R(G^+)$  rappresenta il *pagerank* positivo calcolato come spiegato in precedenza;
- $R((G^\#)^-)$  rappresenta il web ribaltato su cui calcolo la parte negativa (vedremo più avanti come calcolarla).

Quindi: sul web ribaltato calcolo la parte negativa e ottengo l'antimisura. Dalla parte positiva sottraggo questa antimisura. Ottengo la parte infernale  $R^J$  su cui calcolo il *pagerank* negativo come calcolo quello positivo. Un **vantaggio** di questa tecnica di calcolo è una sola misura da ottimizzare perché riutilizziamo la misura che abbiamo già per creare il suo riflesso. Di **contro** però dobbiamo cercare altro spazio informativo da cui attingere per i valori di  $W^+$  e  $W^-$ .

#### 8.2.7.5 Email with web

Quale spazio migliore se non quello che viaggia parallelamente al web? Le email anche se staccate contengono preziose informazioni per il web classico. Inoltre anche lo spazio delle email può essere visto come un'immensa rete:

- Ogni **indirizzo mail** è un nodo.
- Le **mail** inviate invece sono archi.

Proprio come le pagine con i link! Questo è solo una descrizione superficiale, si può andare molto più a fondo e specificare meglio il grafo (per esempio sfruttando anche i contatti di una casella di posta). Arriviamo al passo importante, integriamo il web classico alla rete delle mail. Se una mail contiene un qualche URL ho il collegamento! Viceversa vale lo stesso se una pagina contiene un indirizzo mail. Così si ottiene una super estensione dello spazio web.

**Riempiamo il Janus graph** Usiamo quindi queste nuove informazioni:

##### **Parte positiva:**

- *activeness*: un indirizzo email più attivo dà più valore.
- *history*: una pagina web aggiornata conta di più.
- età: l'età del dominio, applicabile anche alle mail.

**Parte negativa:** ogni volta che una mail è classificata come *spam* lo spazio che la ospita acquisisce valore negativo così come gli URL delle pagine che contiene acquisiranno negatività.

I vantaggi che portano la mail sono:

- È più probabile che siano scritte a mano.
- È più facile tracciare l'informazione che è più pura e autentica.
- È più facile distinguere comportamenti reali da artificiali mentre nel web è difficile il controllo.

Di contro l'analisi è più difficile perché il fattore negativo arriva da flussi di cui non abbiamo niente a che fare: il *pagerank* cambia con eventi legati al solo *spam* delle mail.

**Altre estensioni: SIS** Lo spazio informativo non si ferma alle mail. Si può estendere considerando anche l'identità delle persone: SIS (*Social Information System*). Lo spazio informativo quindi si espande considerando il sistema sociale. Ogni oggetto ha un lato positivo e un lato negativo e anche una persona. Per tenere traccia di questo si utilizza un codice univoco: un UID (*User Identifier*) che corrisponde ad un'identità univoca. Otteniamo così anche un *social rank*.

Ad ogni pagina posso associare anche le persone che ci stanno dietro e scegliere se conta di più un flusso uguale in misura ma che coinvolge molte persone o, viceversa, poche persone. Negli ultimi anni si sta cercando di passare dal *ranking* normale al *social ranking* per avere più controllo e aumentare le contromisure, ad esempio le *spam farm*. Questo è uno dei motivi per cui ci sono sempre più interessi a togliere lo schermo della privacy. Le assunzioni cadono:

- l'unità di una pagina non c'è più e può essere considerata in pezzi diversi dei rispettivi autori.
- la struttura di navigazione del web non ha più cammini uniti ma questi possono rompersi.

#### 8.2.7.6 Email 2.0: andiamo oltre

Non solo la rete delle mail ma in futuro avremo un campo informativo vastissimo, siti social, siti di new, wiki, twitter e tanti altri verranno trattati in modo diversi, spazi informativi che possono unirsi allo spazio del web usando principi sociali integrati nel SE. Un esempio è l'informazione raccolta dai sistemi Android, come l'uso del *Play store*.

## Chapter 9

# Il nome giusto

Nel web, ma anche in generale, la scelta del nome è fondamentale. L’impatto del nome arriva fino al 50% di varianza d’impatto nel pubblico. La media è il 20-30%.

### 9.0.1 Regole d’oro

Di seguito un elenco di regole relative all’inglese per la scelta di un nome giusto:

- I nomi corti funzionano meglio di quelli lunghi.
- Il nome deve essere unico evitando di confondersi con altri nomi.
- Prendersi sempre il dominio *.com*; nella media ha un impatto maggiore del 4,5%.
- Il nome deve essere facile da memorizzare e da scrivere. (questa regola si rivela come un test utile nella scelta tra le alternative individuate)
- Scegliere parole che esistono piuttosto che inventarne di nuove. IN ogni caso mescolare parole conosciute con quelle inventate. L’impatto va da -5% a 1,5%.
- Attenti al suono della parola, deve essere piacevole e armonioso.
  - Se inizia con una vocale ha un impatto maggiore del 3,7%.
  - Se inizia con una semivocale *r, j, y, w* +2,9%.
  - Se inizia con consonante *f, v, s, z* +3,3%.
  - Se inizia con consonante *p, k, t* +1,9%.

I suoni associati alle parole brutte danno svantaggio. In alcuni contesti invece creano effetto positivo (pornografia).

- Niente trattini "-", -3%.
- Molto bene i numeri, +8,2%.

Una nota a parte per il **processo di scelta**. Capita spesso che dopo aver controllato la lista dei nomi trovati liberi il giorno dopo vengano presi da altri. I siti distributori di domini infatti pubblicizzano quei nomi che vanno per la maggiore. Per sicurezza si consiglia l'uso di [www.internic.net](http://www.internic.net).

## 9.1 Indirizzi e nomi dal lato tecnico

Analizziamo l'altra faccia dei nomi: il lato tecnico.

### 9.1.1 URI, URL e URN

Gli URI (*Uniform Resource Identifier*) sono i nomi con cui si identificano le risorse web. L'insieme degli URI comprende anche gli URL e gli URN:

**Ad esempio:**

*www.sito.com* non è un URI poiché non è completo.

*http://www.sito.com* invece è un URI, precisamente un URL.

Attenzione! anche *news:it.cultura* è un URI.

Gli URI si suddividono in:

**URL:** *Uniform Resource Locator*, definiscono come raggiungere una risorsa;

**URN:** *Uniform Resource Name*, identificano il nome di una risorsa e restano unici e persistenti anche quando questa sparisce.

Gli URI possono essere di due tipi:

**Assoluto:** indirizzo completo;

**Relativo:** l'URI diventa assoluto grazie all'informazione derivante dal contesto.

### 9.1.2 Struttura URI

Un URI ha la seguente struttura:

schema: parte-dipendente-dallo-schema

Dove lo schema definisce la semantica (significato) dell'URI che dà significato alla seconda parte.

### 9.1.3 URI gerarchici

La struttura di un URI gerarchico è la seguente:

`schema://authority path ? query`

Dove:

**"//"**: sta per URI gerarchico;

**"authority"**: è la parte che indica chi risponde in caso della risorsa;

**"path"**: è il cammino che può comporsi di 0 o più segmenti della forma  
"/segmento";

**"hash "# "**: serve per indicare sottorisorse all'interno di una stessa risorsa.

**"? query"**: sono i parametri passati e interpretati dalla risorsa.

### 9.1.4 URI opachi: URN

Gli URI opachi hanno la seguente struttura:

`schema : opaque_part`

Dove *"opaque\_part"* son delle specie di cammini ma senza l'uso dello slash *"/"*. Un esempio è: *mailto:director@cnn.com*.

Gli indirizzi URI di tipo opaco sono URN:

`urn : NID : \dots`

Dove NID rappresenta l'identificatore del *namespace*. Un esempio è: *urn:isbn:0-395-36341-1*

### 9.1.5 Oltre gli URI

Gli URI sono con codifica ASCII, per supportare i caratteri non solo latini è nato IRI (International Resource Identifiers).

### 9.1.6 Attacchi omografici

Con l'uso degli IRI si sono però peggiorati gli attacchi omografici, ovvero l'uso di una grafia equivalente per imitare il nome di altri siti (ad esempio G00gle al posto di Google). Si è ricorsi quindi alla validazione con la firma mostrata dai browser.



### 9.1.7 Problema opacità URI

Un problema degli URI è che non danno in nessun modo informazioni sulla risorsa. Per esempio:

*http://www.sito.it/a/pag.html*

Non dice assolutamente che questo sito è in italiano o questa pagina è html. È una stringa opaca, l'unico modo per sapere informazioni sulla risorsa è interrogare l'http.

#### 9.1.7.1 L'idiozia umana

L'idiozia umana però non ha tardato ad arrivare e infatti dalla compagia ICM registry (venditrice di domini) arriva la proposta di usare il TLD (*Top Level Domain* ovvero far sì che il ".qualcosa" avesse un significato per la risorsa che lo contenesse nel proprio nome. Ad esempio:

- .xxx per siti pornografici;
- .kids per i bambini;
- .adult per contenuti per adulti ...

La proposta poteva essere di tipo "leggero" ovvero lasciare che l'uso fosse a scelta o "forte", imporlo per legge. In termini monetari il costo sarebbe stato nullo ma in termini sociali?

per la proposta "leggera" nessuno avrebbe adottato il modello TLD perché sarebbe stato vittima di filtri e quindi avrebbe utilizzato il ".com" per convenienza.

Per la proposta "forte" invece i problemi sarebbero stati ancora maggiori. Cosa definisco pornografico? La foto di Lenna riportata in figura 9.1 è stata usata per anni come immagine sperimentale a scopo illustrativo e grafico finché non fu scoperto derivasse dal giornale di playboy. Da quel giorno fu censurata in ogni dove. Il concetto di pornografico cambia in base all'informazione che c'è dietro.

**L'esempio di Douglas Crackfor e JSMin** Un altro esempio di "ambiguità sociale" proviene dalla libreria JSMin per minimizzare javascript. Nella licenza del software troviamo: *"shall be use for good, not evil"*. Una semplice battuta che però ha generato interessi legali...



Figure 9.1: Il nome giusto - Lenna e l'idiozia umana

**Morale:**

“Usare gli URL per scopi che non sono i suoi è stupidità tecnica e politica.”

...infatti nel 18 marzo 2011 il ".xxx" e altre decine di postfissi sono approvati... E la società ICM registry ha fatturato 200 milioni in più...

## Chapter 10

# L'informazione e il web semantico

### 10.1 La torre di Babele del web

#### 10.1.1 Un esempio di problema informativo

Con l'espansione del web si sperimentano i primi bot cattura informazione. Uno di questi sistemi creato dal MIT permetteva un acquisto in pochi passi. Una *query* famosa fu "manda una rosa alla mia ragazza", il bot trovò il prodotto desiderato e anche ne trovò un altro che corrispondeva a tutt'altro. Lo stesso capitò alla RIA (l'equivalente della SIAE) con un proprio bot che scovò una canzone piratata del cantante Usher ma che in realtà si rivelò una canzone completamente diversa, creata a scopo educativo dal professor Peter Usher. Molti investimenti sono stati fatti e ad oggi i bot sfruttano i tipi semantici, comprendendo. Vediamo come:

#### 10.1.2 Evoluzione

Ma se al posto dell'Html avremmo informazione strutturata? Così facendo potremmo rimuovere le ambiguità e le macchine potrebbero "capire" le informazioni del web. Uno strumento per fare questo può essere l'XML, semplice e flessibile, creato proprio per la necessità di strutturazione dei dati. Purtroppo proprio da questi stessi vantaggi nacquero molti dialetti che resero impossibile l'aggregazione. L'XML uscito fallimentare dall'*open world* portò quindi alla necessità di creare un nuovo livello superiore che permettesse l'aggregazione e un nuovo strumento che lo descrivesse: l'RDF. Vediamo i vantaggi rispetto l'XML:

- le informazioni mappano su un modello non ambiguo. In ogni modello RDF si può riconoscere quali bit rappresentano la semantica e quali la sintassi di contorno.

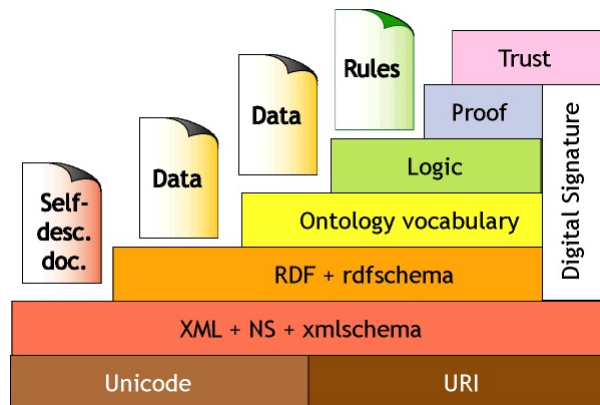


Figure 10.1: Il web semantico - *Layer* che compongono l'informazione

- l'RDF è parte del web semantico.

L'RDF è quello strumento che mancava per rendere possibile l'aggregazione dell'informazione e il ragionamento automatico su di essa.

## 10.2 Il web semantico

### 10.2.1 Babel Fish: l'RDF

Babel Fish è un traduttore istantaneo biologico descritto all'interno del libro *Guida Galattica per gli Autostoppisti* di Douglas Adams. Il modello RDF (*Resource Description Framework*) può certamente interpretare questo ruolo di traduttore dal web che conosciamo a quello semantico come vediamo nella figura 10.1.

#### 10.2.1.1 Il modello RDF

Il modello RDF è un *framework* comune che descrive metadati, relazioni e concetti garantendo l'interoperabilità tra essi. Si basa sulla grammatica di base:

⟨ soggetto - predicato - complemento oggetto ⟩

Questi tre elementi possono contenere due diversi tipi di dati:

- URI,
- Stringhe letterali;

che compongono la struttura linguistica e logica più semplice. Questa struttura può essere visualizzata come un grafo 10.2.

L'RDF può essere rappresentato in due modi differenti:

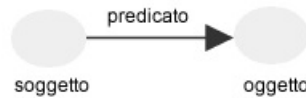


Figure 10.2: Il web semantico - Grammatica di base

**dialetto XML** usando la struttura XML per alzare la complessità e arricchire il significato.

**N-triple** usando le triplette soggetto-predicato-verbo.

Esempio di tripletta della frase *"il cielo è blu"* contenente due stringhe:

```
<rdf:Description rdf:about='cielo'>
  <v:essere>blu</v:essere>
</rdf:Description>
```

L'RDF è un linguaggio separato dal web comune, per integrarlo si è ricorso ad una versione XHTML2: l'RDFa che aggiunge gli attributi *about* (il soggetto) e *property* (il verbo), il contenuto è il complemento oggetto.

#### 10.2.1.2 Funzionalità e proprietà di RDF

**Aggregazione:** l'unione di due grafi di conoscenza è un grafo creato tramite collegamenti automatici dei nomi del web (URI). Questa è la proprietà più importante e che decreta il successo di RDF rispetto ai dialetti XML.

**Contenitori:** concetti logici di AND e OR.

**Variabili:** oggetti logici non specificati.

**Monotonicità:** preso un grafo, se l'informazione espressa in esso supponiamo sia vera allora tutti i sottografi sono veri.

**Reificazione:** garantisce la monotonicità, riduce ad un oggetto (*"cosifica"* cit.) una asserzione. Per esempio nella frase *"Gromit dice che 'la luna è di formaggio'"*, la frase nel sottolivello *"la luna è di formaggio"* è in realtà un'altra frase che grazie al processo di reificazione diventa un oggetto, per cui, anche se non sappiamo sia vero ciò non influisce nella veridicità della prima frase. Grazie a ciò abbiamo anche una diversificazione di livelli.

Gli oggetti devono essere anche classificati garantendo un costo computazionale minimo. Abbiamo bisogno di un **sistema di classificazione dell'informazione**: un'ontologia che contenga tipi semantici. Mentre i **tipi** danno il formato sintattico dell'oggetto, i **tipi semantici** gli danno il significato astruendo la rappresentazione sintattica.

### 10.2.1.3 I tipi semantici

Abbiamo visto con il caso "*Usher vs Peter Usher*" che nel web mancano i tipi semantici. I tipi semantici possono essere astratti come classi. Un insieme di classi forma un'ontologia. Un'ontologia è costituita da:

**Etichette:** rappresentanti caratteristiche informative (possono essere più di una per oggetto).

**Struttura:** rappresentante la gerarchia di tutte le classi. Più una struttura è ricca e più l'ontologia mi permette di espandere le cose e la comprensione di esse. È grazie alla struttura che è possibile fare **controlli di integrità semantica** e **ragionamenti**.

### 10.2.1.4 Caratteristiche di RDF Schema

L'RDF offre uno schema con una struttura informativa fatta di:

- *Class*;
- *subClassOf*;
- *individual*.

Questo per gli oggetti mentre per i verbi abbiamo a disposizione:

- *Property*; (la relazione, il verbo)
- *subPropertyOf*;
- *Domain*;
- *range*. (dove la proprietà è applicata)

Grazie all'RDF Schema, quindi, possiamo **categorizzare l'informazione**.

### 10.2.1.5 Oltre l'RDF Schema

Abbiamo fissato delle regole semantiche ma siamo ancora al livello base delle ontologie (stato tassonomie). SI può fare molto di più: l'RDF permette l'aggregazione automatica attraverso l'URI. Ma gli URI presentano problemi di ambiguità.

## 10.2.2 L'architettura del web (Tim Berners-Lee)

Tim Berners-Lee prevedeva già questo alla nascita del web e infatti il web è nato seguendo questi assiomi:

- 0a) A qualsiasi risorsa dovunque sia, essa può avere un nome. (*assioma di universalità*)

- 0b) Ogni cosa di significato dovrebbe avere un URI. (*assioma di universalità 2*)
- 1) Non importa a chi o dove specifichi un URI, ma che abbia sempre lo stesso significato. (*global scope*)

### 10.2.2.1 Dare un nome ad ogni cosa: il problema degli URI

Trovare un nome (web) non è banale:

**URI *variant problem*:** esistono molte varianti per o stesso concetto.

**URI *variant law*:** l'utilità decresce esponenzialmente con il numero di varianti (legge della varianza degli URI). Più un nome ha diversi significati più questo perde valore.

**URI *variant size*:** un URI può acquisire diversi significati in base al dominio di interesse. "Problema *big red barn*" cit. (se un bambino indica una mucca illustrata in un libro, ciò che indica è "una mucca" o "un libro"? Esistono livelli diversi).

### 10.2.2.2 La soluzione

La soluzione a questo inghippo è aggiungere più informazione. Abbiamo bisogno di maggiore informazione, un nuovo *layer*: lo strato ontologico e un nuovo linguaggio.

### 10.2.3 Strato ontologico: OWL

Il supporto di base fornito da RDF Schema è stato quindi esteso con lo strato ontologico. Questo strato è descritto attraverso il linguaggio OWL (*Web Ontology Language*) che si occupa di collegare e relazionare i vocaboli dando ad ognuno il proprio dominio d'interesse (la propria ontologia appunto).

#### 10.2.3.1 Caratteristiche e proprietà di OWL

In OWL si può parlare dell'(in)uguaglianza degli oggetti. Per fare questo definisce le proprietà:

**EquivalentClass:**

**equivalentProperty**

**sakIndividualeAS**

**differentFrom:**

**allDifferent**

## SameIndividualAs

Grazie a queste proprietà si riesce a risolvere il problema dei nome. Altre funzionalità disposte da OWL:

- `inverseOf`: collegare due proprietà per complementarità.
- `transitiveProperty`: capire se una proprietà è transitiva.
- `symmetricProperty` (l'amicizia su Facebook è simmetrica).
- `inverseFunctionalProperty`: specificare se un certo verbo è funzionale o no (dato un oggetto c'è un solo complemento oggetto associato).

Sono inoltre possibili utilizzare restrizioni sulle proprietà delle classi:

- `allValueFrom` ( $\forall$ );
- `someValueFrom` ( $\exists$ );
- `minCardinality`
- `maxCardinality`
- `cardinality`

### 10.2.3.2 Il problema: ragionarci

Abbiamo tutte queste informazioni su cui ragionarci. Ma come? Purtroppo quando si passa ad un ragionamento ad alto livello la logica diventa indecidibile. Per descrivere espressioni ad alto livello sulle macchine è necessaria una traduzione a basso livello poiché la logica di prim'ordine, che fa uso dei costrutti logici  $\exists$  e  $\forall$ , non è decidibile (la macchina non sa creare un programma che termini sempre). È il motivo per cui non abbiamo ancora linguaggi di programmazione ad alto livello che seguono una scrittura più vicina al nostro linguaggio.

### 10.2.4 SPARQL

Abbiamo bisogno di un linguaggio che permetta una comunicazione di alto livello (relazionale) e che si occupi lui di definire la parte di basso livello. Un linguaggio simile funzionante è SQL in cui definisce tabelle e relazioni. Il vincoli di SQL è che alcune cose sono limitate, questo per permettere di garantire la terminazione del programma. La stessa cosa è stata applicata al web semantico sostituendo le relazioni tra tabelle con relazioni in un grafo generico. Nasce così SPARQL (*SPARQL Protocol And RDF Query Language*, il linguaggio per permettere le *query* nel web semantico seguendo le scelte di design di SQL).



#### 10.2.4.1 Modellazione con SPARQL

Il fattore principale di SPARQL è il *pattern matching* tra triplette nella struttura a grafo.

La struttura di una *query* SPARQL è:

```
PREFIX ...  
SELECT ...  
FROM ...  
WHERE ( ...  
)  
ORDER BY ...
```

#### 10.2.5 Vocabolari ontologici

##### 10.2.5.1 DC: Dublin Core

Uno dei primi tentativi di strutturare il web in maniera semantica. Definisce le proprietà di base fondamentali e essenziali che modellano l'intera struttura dei dati semantici. Queste proprietà di base individuate sono 15 che elenchiamo per completezza:

1. *Title*
2. *Creator*
3. *Subject*
4. *Description*
5. *Publisher*
6. *Contributor*
7. *Date*
8. *Type*
9. *Format*
10. *Identifier*
11. *Source*
12. *Language*
13. *Relation*
14. *Coverage*
15. *Rights*

Un insieme compatto che mi permette di definire una pagina web in triplette che compongono un grafico della conoscenza su cui è possibile effettuare delle *query*. Per esempio, usando i *tag meta* è possibile tramite la *keyword* 'DC' mettere a disposizione queste triplette direttamente nel web in modo che bot intelligenti ricavino informazioni utili.

```
<meta name="DC.title" content="How to use DC" />
<meta name="DC.creator" ... />
...
```

#### 10.2.5.2 FOAF: Friend Of A Friend

Altro vocabolario per altro usato da Google. Nato per la socialità e non le pagine web. Definisce le proprietà base di una persona. Per la class Person per esempio abbiamo:

- *Name* (pagina web)
- *Title* (pagina web)
- *Firstname*
- ...
- *myerBriggs* (uno dei 16 tipi di personalità)
- ...

Ad esempio:

```
<foaf:Person>
<foaf:name>Massimo Marchiori</foaf:name>
<foaf:mbox rdf:resource=mailto:massimo@w3.org />
</foaf:Person>
```

È inoltre possibile esprimere il concetto di 'amicizia' con la relazione `foaf:Knows`. Qui è possibile osservare la difficoltà della scrittura ad annidamento di XML a differenza delle semplici triplette.

#### 10.2.6 Un esempio

PlanetRDF è un esempio di base di dati costruita sul web semantico. Tramite le informazioni semantiche con i vocabolari descritti sopra in ogni pagina è stato creato un grafo della conoscenza su cui era possibile effettuare *query*.

### 10.2.7 Ancora SPARQL

Preso il grafo della conoscenza, una base di dati, con SPARQL è possibile eseguire la propria *query*. Per esempio vogliamo prendere l'indirizzo del blog di 'Mario Rossi' dal database planetRDF.

```
PREFIX foaf:<http://xmlns.com/foaf/0.1/>
SELECT ?url
FROM <http://planetRDF.com />
WHERE (
  ?someone foaf:name "Mario Rossi"
  ?someone foaf:webblog ?url )
```

#### 10.2.7.1 Extra SPARQL

Una caratteristica di SPARQL interessante è la possibilità di ricerca con dati opzionali, ossia nulli o assenti. Nel web semantico è facile avere informazioni parziali per questo SPARQL è stato previsto l'operatore **OPTIONAL**. Ad esempio:

```
PREFIX foaf:<http://xmlns.com/foaf/0.1/>
SELECT ?name ?picture
FROM <http://planetRDF.com />
WHERE (
  ?someone foaf:name ?name
  OPTIONAL{
    ?picture foaf:picture
  }
)
```

La *query* cerca il nome e la foto di qualcuno, grazie all'uso di **OPTIONAL**, nel caso in cui qualcuno non abbia caricato la propria foto questa *query* non darà errore.

### 10.2.8 Un problema di complessità

SPARQL è decidibile come SQL? È efficiente? Qual è la sua complessità? La **gerarchia dei problemi**, individuata dagli studi sulla complessità, è:

$$NL \subset P \subset NP \subset PSPACE \subset EXPTIME \subset EXPSPACE$$

Nel design del linguaggio SPARQL si sono seguite le scelte di SQL, per cui è decidibile (ovvero termina) ma ha una complessità *PSPACE* (spazio polinomiale a rischio).

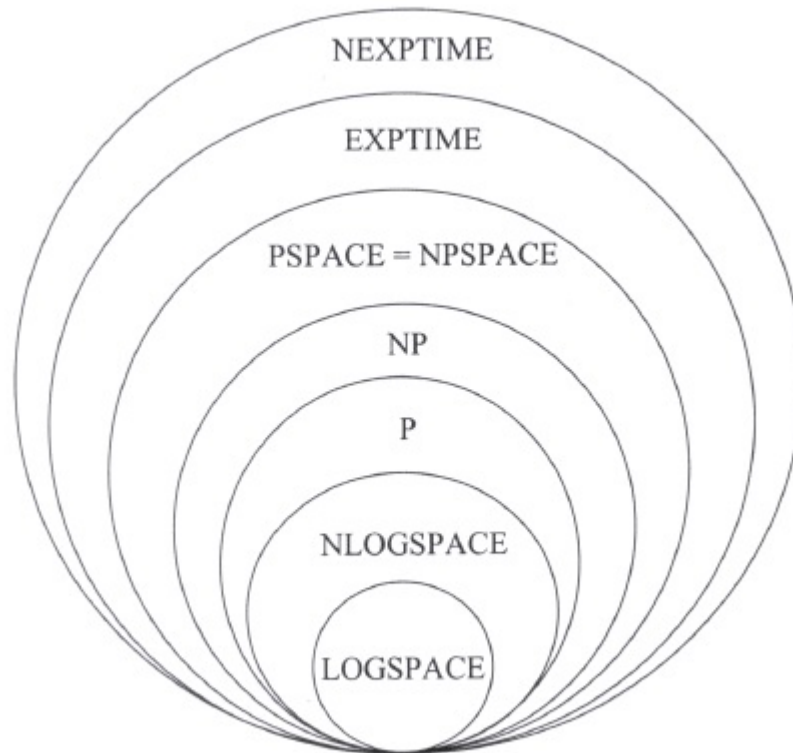


Figure 10.3: L'informazione e il web semantico - Gerarchia dei problemi

#### 10.2.8.1 La soluzione di OWL

Con OWL invece si è optato per mantenere

sia l'**espressività**, avere una logica decidibile ma limitata,

e sia la **computabilità**, avere una logica espressiva ma complessa da gestire.

Per garantirle entrambe, si sono creati dei sottoinsiemi di OWL:

**OWL *lite***: decidibile con uso della logica SHIFT.

**OWL *DL***: decidibile con logiche descrittive SHOIN.

**OWL *full***: indecidibile ma sfrutta le logiche più avanzate.

### 10.2.8.2 Complessità e statistica

La complessità della logica SHIFT è *EXPTIME* (*query* a tempo esponenziale) mentre per SHOIN è *NEXPTIME* (tempo esponenziale non deterministico) mentre per OWL *full* non possiamo dire che termini sempre. Viene da pensare per quale assurdo motivo usare strumenti con questa complessità in zone esponenziale ma la complessità è una statistica. È molto più rilevante la statistica su dati non casuali e soprattutto non su qualsiasi combinazione esistente. È molto più rilevante la media ricavata degli input d'interesse.

**Complessità e pratica** Nell'uso pratico anche se la complessità è in tempo esponenziale la maggior parte delle *query* rientra in complessità molto più basse. Per esempio molta dell'algoritmica di Google sta in *EXPTIME* ma le *query* più frequenti stanno molto al di sotto. Solo alcune *query* sono esponenziali e il problema è risolto attraverso interruzioni di esse con timer. In SPARQL che è in *PSPACE* soltanto togliendo la *keyword* **OPTIONAL** scendiamo nella complessità (assoluta) in *co - NP*. Anche OWL DL e OWL *Lite* a livello statistico molta informazione all'interno del web non è casuale ma segue un ordine e non rientra nella logica indecidibile ma in logiche studiate: ALC e AL che corrispondono rispettivamente a complessità *PSPACE* e *P*.

## 10.3 Il web semantico diventa Linked Data

Passiamo all'uso del web semantico che per una scelta di *marketing* è stato definito come **Linked Data**.

### 10.3.1 Classificazione LOD

I LOD sono i tipi di *Linked Data Open*, ossia disponibili a tutti gratuitamente. Sono classificati in una scala da 1 a 5 stelle.

★ dati sul web liberi (*open*).

★★ dati sul web liberi con formato strutturato ovvero *machine readable*.

★★★ dati sul web liberi con formato dati non proprietario (non necessariamente con RDF).

★★★★ dati sul web liberi con formato *semantic web*.

★★★★★ dati sul web liberi con formato *semantic web* collegati ad altri per darli un contesto.

Usher Official Website | Latest Usher News & Media  
[ushervworld.com/](http://ushervworld.com/) Traduci questa pagina  
 Usher's Official Website - The largest compilation of media, photos, videos, news & more. Read about Usher's latest event and tour information.

Usher - Wikipedia  
<https://it.wikipedia.org/wiki/Usher>  
 Oltre alla musica Usher ha intrapreso anche altre attività: ha creato la sua etichetta musicale, la US Records, ed è uno dei proprietari della celebre franchigia ...  
[Confessions](#) - [Album](#) - [My Way](#) - [Live](#)

Usher (album) - Wikipedia  
[https://it.wikipedia.org/wiki/Usher\\_\(album\)](https://it.wikipedia.org/wiki/Usher_(album))  
 Usher (album). Da Wikipedia, l'enciclopedia libera. Vai a: navigazione ... Usher è l'album di debutto del cantante R&B Statunitense Usher, pubblicato nel 1994.

Usher (singer) - Wikipedia, the free encyclopedia  
[https://en.wikipedia.org/wiki/Usher\\_\(singer\)](https://en.wikipedia.org/wiki/Usher_(singer)) Traduci questa pagina  
 Usher Terry Raymond IV (born October 14, 1978) known mononymously as Usher, is an American singer, songwriter, dancer, and actor. Born in Dallas, but ...

Usher - Yeah! ft. Lil Jon, Ludacris - YouTube  
<https://www.youtube.com/watch?v=GxBSyx85Kp8>  
 17 apr 2015 - Caricato da Usher/VEVO  
 Confessions is Usher's fourth studio album, reased in 2004 by Arista Records. Confessions debuted at ...

Usher on Spotify  
<https://open.spotify.com/artist/23zg3TcAtfWQy7J6upgbUnj>  
 One of the most successful pop-R&B artists of the late 20th century and the early 21st century, Usher debuted during the mid-90s as a fresh-faced teenager.

Usher Raymond IV (@Usher) | Twitter  
<https://twitter.com/usher> Traduci questa pagina  
 The latest Tweets from Usher Raymond IV (@Usher): I had in the ass of a married

**Usher**  
 Cantante

Disponibile su

- YouTube
- Deezer
- Spotify

Usher Raymond IV, conosciuto semplicemente come Usher, è un cantante, ballerino, attore pittore statunitense. È diventato popolare negli anni novanta come performer R&B, pubblicando gli album My Way nel 1997 e 8701 nel 2001. [Wikipedia](#)

Altezza: 1,73 m

Film e programmi televisivi: The Voice, The Faculty, altri  
 Coniuge: Grace Miguel (s. 2015), Tameka Foster (s. 2007-2009)

Canzoni

I Don't Mind	2014
DJ Got Us Fallin in Love	2010

Raymond v. Raymond

Figure 10.4: Il web semantico - Informazione strutturata (sulla destra) raccolta tramite *lifting*

### 10.3.2 Lifting & lowering

*Lifting* & *lowering* corrispondono, rispettivamente, al passaggio dal mondo a 3 stelle al mondo semantico di 4 e 5 stelle e viceversa. Esempi di strumenti di *lifting* sono:

- DR2Q: trasforma il linguaggio SQL in RDF.
- Triplify: *plug-in* leggero che crea una struttura semantica da database relazionali.
- Openlink 'virtuoso': server universale che permette la relazione tra SQL, XML e RDF.

Per *lifting* dal formato più basso:

- Open Calais: da un significato al testo e all'informazione pura.
- Spotlight.
- Alchemy.

Un esempio di quanto potenziale offrono questi strumenti ce lo fornisce il sito [www.wikido.com](http://www.wikido.com). Questo sito preleva da un insieme di siti selezionati informazione che con Open Calais viene trasformata in informazione di alto livello ottenendo dati strutturati. Lo stesso fa Google con particolari *query* (vedi figura 10.4).

### 10.3.3 Connettere le informazioni: il Mash Up

Abbiamo visto le immense potenzialità che derivano dal *lifting* dei dati, ma come è possibile fare questo? Di seguito spieghiamo un tipo di algoritmo adottato da Google che permette questa "magia".

**Il problema:** Dati due grafi di conoscenza dobbiamo collegarli (se sono collegabili). Per fare questo ci serve una funzione che dati due oggetti web in input, abbia in output un valore di similitudine. Immaginando una funzione  $simil(x, y)$  posso ottenere un livello di similitudine, se ottengo entro una certa soglia relazione  $x$  con  $y$ . Con una tecnica di *brute force* otterrei una complessità pari a:

$$O(|G_1| \cdot |G_2|)$$

perché sto elaborando i *BIG DATA*.

Una tecnica più efficace sfrutta la **distanza di Levenshtein** che misura la distanza tra due stringhe. Ad esempio una stringa con un carattere diverso da un'altra ha distanza 1. È grazie all'uso di esse che Google o qualsiasi altro motore di ricerca mette a disposizione una correzione della *query* inserita dall'utente, ricerche correlate o suggerimenti.

Andando più fini, possiamo sfruttare le proprietà delle distanze:

- $d(x, y) \geq 0$
- $d(x, y) = 0 \Leftrightarrow x = y$
- $d(x, y) = d(y, x)$
- $d(x, y) \leq d(x, z) + d(z, y)$  (limite superiore)

Ci interessa la distanza tra due oggetti e con l'ultima proprietà posso ottenere un'approssimazione:

$$d(x, z) \leq d(x, y) + d(y, z) \quad (10.1)$$

$$d(x, z) - d(y, z) \leq d(x, y) \quad (\text{limite inferiore}) \quad (10.2)$$

Da cui:

$$d(x, z) - d(y, z) \leq d(x, y) \leq d(x, z) + d(z, y)$$

Così, invece di calcolare  $d(x, y)$  calcoliamo un range d'approssimazione se sappiamo la distanza da un punto  $z$  (che descrive geometricamente una stringa). Per sfruttare ciò quindi abbiamo necessità di calcolare la distanza dai punti  $z$  (*exemplar*). Con più punti *exemplar* più avremo un range preciso al costo di più calcoli. Fissiamo quindi un numero di punti *exemplar* e calcoliamo solo in essi la distanza riducendo il numero di calcoli. Per diminuire al minimo il numero di punti *exemplar* e far sì che questi diano un range d'approssimazione accettabile dobbiamo prendere punti distribuiti

equamente. Iniziamo scegliendo un punto e calcoliamo tutte le distanze tra lui e gli altri punti. Dopodiché scegliamo il punto con la distanza maggiore calcolata e ripetiamo lo stesso procedimento in modo da ottenere sempre i punti il più distante possibile. Decisi quanti sceglierne otteniamo uno spazio informativo diviso in aree associabili ai punti focali. In ogni zona posso avere delle classifiche dei punti più vicini rispetto ai punti focali, per cui, tornando al problema iniziale, utilizziamo questa tecnica per ottenere i valori di similitudine fissando sempre una soglia e utilizzando i range di approssimazione. Il calcolo:

$$simil(x, y) \leq soglia$$

Con questa tecnica diviene:

$$Sed(x, z) - d(y, z) > soglia \Rightarrow d(x, y) > soglia$$

Con un'unica operazione (una differenza) sappiamo già quale punti scartare poiché le distanze le abbiamo già calcolate.

Ad esempio: Prendiamo un *exemplar* del grafo  $G_1$  e scegliamo un elemento di  $G_2$ :  $y$ , non appena:

$$simil(x, exemplar) - simil(x, exemplar) > soglia \quad (10.3)$$

$$\Rightarrow simil(x, y) > soglia \quad (10.4)$$

e scartiamo  $y$ . Inoltre anche per ogni altro  $x$  della partizione (sotto la classifica della zona precalcolata) varrà la stessa cosa sempre (la soglia sarà ancora maggiore) per cui rimuoviamo il calcolo per tutti gli altri punti al di sotto della classifica, con un'operazione soltanto. Solo per le coppie non scartate da questa tecnica facciamo  $simil(x, y) > soglia$ .

### 10.3.3.1 Quanti punti scegliere?

Scegliere il numero di punti *exemplar* si rivela il vero problema. Si è visto sperimentalmente che il numero ottimale di punti è la radice quadrata della dimensione dei dati (grafo più grande). La complessità quindi risulta:

$$O(|E| \cdot |G_1|) + O(|G_1| \cdot |G_2|) = O((|E| + |G_2|) \cdot |G_1|)$$

Siamo partiti da con una complessità pari a  $O(|G_1| \cdot |G_2|)$  con la tecnica *brute force* e siamo arrivati a  $O((|E| + |G_2|) \cdot |G_1|)$ . Abbiamo ottenuto un algoritmo peggiore ma non nella pratica, perché questo algoritmo messo in pratica funziona splendidamente e consente di avere un risparmio di costo computazionale del 95% rispetto al *brute force*.

### 10.3.3.2 Esporre i dati

Per esporre i dati nel web abbiamo diversi modi. Uno può essere quello di mostrare i dati come file sotto un URI, ma questo è poco pratico. Nella pratica si preferisce ricorrere a interfacce con cui fare *query* usando SPARQL,



offrendo in questo modo un servizio web molto potente e flessibile. Questi *query* fatte attraverso indirizzo web con classici "*urlencoding*" e passaggio parametri con GET HTTP sono chiamate ***endpoint***. Un esempio di SPARQL *endpoint*:

```
PREFIX dc <http://purl.org/dc/elements/1.1/>
SELECT ?book ?who
WHERE ( ?book dc:creator ?who )
```

Un esempio concreto di un tale servizio è [dbpedia.com](http://dbpedia.com) da cui è possibile effettuare *query* SPARQL e ottenere pagine come questa: tutta la conoscenza di [Albert Einstein](#) in una pagina. Le ontologie principale sono date da [schema.org](http://schema.org). Un altro sito interessante è [www.visualdatamedia.org](http://www.visualdatamedia.org), il quale mette a disposizione un servizio che permette di navigare il grafo delle conoscenza sulle parole date.

## Chapter 11

# Mobile Web (e App)

Solo recentemente i dispositivi mobile hanno spopolato e la tecnologia ha superato di gran lunga i web designer che sono impreparati nell'ambito mobile emergente. Nel 2013 l'accesso a internet da dispositivi mobile ha superato quello di desktop e laptop ed è in costante crescita. Nonostante questo trend, 530 siti nella top 1000 del mondo non dispongono di una versione mobile e il 25% di questi sfiora lo schermo.

### 11.1 Un po' di storia

Nel marzo 2013 avviene un'importante scelta aziendale in Google. Il team di sviluppo di Android che non portava risultati soddisfacenti è inglobato dal team di Chrome che invece aveva successo. L'idea era ed è quella di avere convergenza tra mondo mobile e web.

Già prima si era cercato di percorrere questa rotta da Google ma le cose non andarono bene visti i contrasti con Apple che non voleva collaborare. Si pensi che lo stesso Steve Jobs era contrario alle app. Da qui il motivo dell'acquisto del sistema Android da parte di Google.

Ora il percorso è ben delineato: la nascita delle hybrid apps scritte usando HTML5 e multiplatforma segnano ancora più visivamente la convergenza tra mobile e web.

### 11.2 Le App

Nascono dalla necessità di minimizzare ancora una volta lo sforzo computazionale delle persone. L'app minimizza enormemente il tempo di accesso al servizio richiesto dagli utenti. Di conseguenza questo porta a maggiori esigenze da parte degli utenti e riduce i tempi di soddisfazioni.

### 11.2.1 Parlano le statistiche

Dalle statistiche emerge:

- quasi un quarto degli utenti usano app più di 60 volte al giorno
- e questo cresce ogni anno del 123%!
- La fascia d'età che meno 'drogata' di app si trova tra i 25 e 35 anni (i motivi sembrano principalmente per la mancanza di tempo).

Le app vincono sul mobile web, gli utenti smartphone passano in media l'84% di tempo giornaliero sulle app e solo il 14% sul web vero e proprio. Nella pratica si capisce il perché:

- il 32% di questo tempo è speso in **giochi** (non sorprende quindi la scelta del nome Google Play per lo store di Google).
- il 28% sui **social**, il 17% è Facebook!

Da notare che tutto questo uso di app (giochi a parte) è solo fruizione di contenuti nel web tramite l'app apposita.

### 11.2.2 L'arena delle App

Quando le statistiche parlano chiaro e muovono un sacco di persone si muovono anche un sacco di soldi e ricerca di successo. È per questo motivo che nel mercato delle app, attualmente, c'è un'enorme competizione:

- Le app hanno vita media bassissima: dai **4 mesi** ad **1 anno**.
  - i game hanno vita media di soli **4 mesi**.
- Se un app resiste ed è ancora in crescita dopo 3 mesi avrà una vita lunga altrimenti è defunta e da considerarsi un insuccesso.

### 11.2.3 La sequenza della morte

Di seguito quella che viene chiamata la *sequenza della morte* di una app descrive al meglio quello già descritto sopra, riportano i dati del comportamento degli utenti di fronte ad un'app.

- il **26%** delle app è aperta al massimo **una volta**.
- il **13%** sono aperte al massimo **2 volte**.
- il **9%** sono aperte al massimo **3 volte**.
- il **50%** degli utenti apre le app al massimo 3 volte e poi basta.

#### 11.2.4 Alla ricerca dell'App

Tanta competizione e tante app defunte in pochissimo tempo. Ma come trovare queste app? È qui che il paragone con i siti web è possibile. Come per essi esistono i motori di ricerca anche per le app esistono questi: gli store. Anche qui infatti si presenta il problema di essere trovati ai primi posti della ricerca nello store proprio come per i siti internet. Per fare ciò è nata l'ASO.

#### 11.2.5 ASO: App search optimization

È il corrispondente CEO per le app e presenta di fatto delle somiglianze prima su tutte funziona per *keywords* che richiede quindi sforzo per un'**ottimizzazione testuale** sui pochi luoghi disponibili nello store.

- Descrizione app.
- Spazio apposito per le keyword.
- Nome dell'app (corrisponde al nome del sito vedere indice NOMI).

Poichè non si possono utilizzare tecniche ipertestuali i motori di ricerca degli store applicano l'uso dei dati del sistema sociale complessivo (SIS) che si basa su quanto segue:

- numero di download (integrati nel tempo).
- tempo d'uso dell'app.
- *ratings e review*.
- disinstallazioni.
- brand.
- metriche di motori di ricerca del web. Per esempio su Google Play sono integrate tutte le metriche positive e negative raccolte sul web per quell'app.

### 11.3 Usabilità: mobile e desktop

Per valutare se una pagina è corretta per dispositivi mobile esistono potenti strumenti. Prima fra tutti il *Google mobile compatibility test*. Esso verifica che siano rispettate alcune caratteristiche che possiamo catalogare in tre componenti base:

1. Essere mobile.
2. Taglia dello schermo.
3. Interazione.

### 11.3.1 L'esempio di Facebook

Una considerazione è doverosa farla sui diversi tipi di device oggi in commercio. Oltre a diverse composizioni hardware abbiamo diverse funzionalità offerte dagli telefoni cellulari. Bisogna porre attenzione al target di riferimento, si pensi ad esempio che non tutti i telefoni hanno il touch. L'esempio del social network mondiale Facebook è esplicativo del problema. Facebook per risolvere questi problemi infatti offre addirittura 3 versioni mobile del sito *facebook.com*:

**m.facebook:** versione per cellulari non touch.

**touch.facebook:** versione per cellulari touch.

**0.facebook:** versione a banda ultra ridotta offerto gratuitamente in tutte le zone dove le reti telefoniche sono lente (fidelizzazione globale dei clienti).

### 11.3.2 Essere Mobile

Essere mobile significa avere un diverso collegamento alla rete: la rete mobile con tutte le conseguenze ovvie. Le connessioni 3G in media sono il 40% più lente delle normali connessioni questo significa che ogni sito web mobile sarà caricato con il 40% in più di tempo. Un disastro se pensiamo ai già discussi timer dell'utente. Fortunatamente con le nuove tecnologie per la rete mobile, il 4G/LTE abbiamo reti in media il 12% più lente.

#### 11.3.2.1 Timer su mobile

Abbiamo visto che i timer causa connessioni di rete mobili si sono allungati del 40%, ma cosa ancora peggiore ad ogni pagina/click l'utente accumulerà un ritardo del 40%. Per far fronte a questo problema e ridurre un po' i timer bisogna ridurre il più possibile il carico delle pagine (0.facebook.com).

- Nel caso **desktop** l'utente aspetta **al massimo 2 secondi** prima che inizini le brutte sensazioni.
- Nel caso **mobile** abbiamo la stessa identica cosa!

*"Non basta cambiare il layout per supportare il mobile."*

**Responsiveness** Lo stesso discorso vale anche per le app, l'azione richiesta dall'utente non deve metterci più di 2 secondi. Si deve seguire il principio della *responsiveness*: non si deve mai far percepire il ritardo agli utenti se non in casi speciali segnalati all'utente.

### 11.3.2.2 Alla ricerca di soluzioni

**Progress bar e spinner:** visto questo inghippo potremmo usare qualcosa per allietare il ritardo inevitabile attraverso tecniche già usate dal lato desktop come *progress bar* e *Spinner*. NO! In qualsiasi caso, anche su desktop, tecniche del genere sono risultate spiacevoli per l'utente. L'effetto è come quello di essere in coda e avere una voce che costantemente ti ricorda di esserlo.

**Transitionig:** tecnica più apprezzata rispetto le precedenti che si propone di tenere impegnato l'utente con un'animazione. Un esempio possiamo trovarlo dal vecchio Netscape che adoperava questa tecnica nel caricamento delle pagine (*skeleton screen*). Esse infatti venivano generate e mostrate man mano che venivano scaricati i dati completamente.

**Preemptiveness:** tecnica che consiste nel far fare qualcosa preventivamente all'utente. Si prenda per esempio l'upload di foto di *Whats App*, l'utente è intrattenuto da una schermata dove viene richiesto un commento testuale prima di inviare il messaggio. In realtà l'app sta utilizzando quel tempo per caricare la foto. Foto caricata, nessuna apparente attesa, utente contento.

### 11.3.3 Taglia dello schermo

Un'altra caratteristica fondamentale del mobile che impatta enormemente sull'usabilità è la taglia dello schermo. Una pagina classica farà fatica ad evitare lo scroll. Abbiamo visto gli effetti dello scroll su desktop, ma su mobile?

- Lo **scroll orizzontale** resta il **male del male**.
- Lo **scroll verticale** non è così male come lato desktop.

#### 11.3.3.1 Scroll verticale su mobile

- lo sforzo fisico e mentale è minimo a differenza del desktop.
- ma risulta deleterio per mostrare scelte quali possono essere liste di prodotti, perché richiede uno sforzo di memoria.

Per guadagnare un po' di spazio e ridurre lo scroll:

- Nelle scelte si evita del tutto l'uso di immagini, restringerle non è cosa gradita.
- Utilizzare le icone al posto del testo, attenzione però a rispettare:  
**explainability:** fornire informazioni testuali se si posiziona il cursore.

**escapability:** possibilità di evitare l'azione se ho già premuto ma non rilasciato.

Una nota per l'uso delle icone. Si ricorda che gli utenti preferiscono **sempre** il testo (vedi confronto tra web e giornali). Si pensi che per abituare gli utenti all'uso dell'icona hamburger, introdotta da Google, sia Chrome che Firefox (finanziato da Google ricordiamo), entrambi browser desktop, l'hanno utilizzata per rappresentare il menu. Questo ha aumentato l'insoddisfazione degli utenti ma nel lungo periodo abituerà essi al suo uso.

#### 11.3.3.2 Invasività - pubblicità

Lo schermo è piccolo e quindi lo spazio per l'odiata pubblicità?

**Pubblicità fissa** Per essa l'ente IAB (*Interactive Advertising Bureau*) ha fissato alcune misure:

**Medium:** 300x250 (per smartphone)

**Full size:** 486x60 (per tablet)

**Leaderboard:** 728x90

Esiste poi l'**interstitial ads** che è la pubblicità che prende tutto lo schermo del cellulare.

**Pubblicità dinamica** Due tipologie:

**Smart banners:** banner con altezza fissata ma ampiezza variabile in base a quello dello schermo. Possono essere **non "scrollabili"** ("orrore!" cit.) e seguono le stesse regole dei banner per desktop.

**Smart app banners:** pubblicità dell'app sul proprio sito. NO! Sono odiate dagli utenti perché considerati veri e propri pop-up.

#### 11.3.4 Interazione: le dita

Un'altra caratteristica dei device mobile è l'assenza del mouse e l'uso delle dita (nel touch). Rispetto al mouse quindi abbiamo un puntatore grezzo definito *fat finger*. Vediamo il perché con alcuni numeri sulla dimensione dei nostri polpastrelli:

- dito medio: 11 mm (di un bambino: 8 mm).
- dito più grande (il pollice): 19 mm.

Da qui conseguono importanti informazioni:

- Un'area cliccabile deve essere grande a sufficienza.
- La **taglia minima** è di **7x7 mm** e zona padding di 2 mm.
- Una **taglia soddisfacente** è **9x9 mm**.
- Seguire il **reversibility principle**, ossia l'azione deve essere reversibile se ho il rischio di sbagliare.

#### 11.3.4.1 Fitts, il ritorno

Non dimentichiamoci della formula di Fitts. In mobile non vale molto come su desktop. Questo perché la taglia dell'oggetto conta ma conta anche la precisione delle dita e le distanze non possono essere calcolate perché dipendono dalla presa del device. Esistono 5 modi più comuni per usare uno smartphone:

- Una mano e uso del pollice come puntatore.
- Una mano tiene il device, l'indice dell'altra è il puntatore.
- Due mani con i pollici come puntatori.
- Le primi due per i mancini.

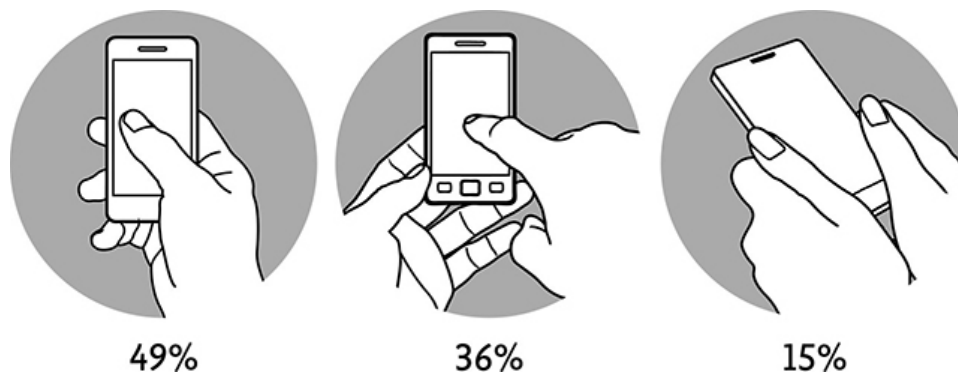


Figure 11.1: Mobile Web - Tipi di impugnatura e percentuale di utilizzo

Come si vede dalla figura 11.1, l'uso del pollice è preferito (75%) e questo garantisce una **pessima precisione**. Ci sono poi delle zone di **bassa usabilità** perché raggiungibili solo allungando la mano, nel caso dei tablet peggio ancora, per questo i controlli dovrebbero essere sempre nella parte inferiore (come i controlli standard degli smartphone). Bisogna poi tenere conto che la forma dello schermo può cambiare da normale a landscape. La migliore interfaccia quindi deve tenere conto di tutti i casi e lasciare la possibilità all'utente di cambiare interfaccia.



**Zone magiche** Riguardo le così definite *zone magiche* su mobile non disponiamo di nessuna finestra. I *fan menu* funzionano molto bene meno invece i *pie menu* perché le dita coprono parti di schermo e quindi anche pulsanti.

## Chapter 12

# Il social web (\*)

### 12.1 Storia

### 12.2 Visibilità

#### 12.2.1 Regole virali

*Emotion*

*Story telling*

*Practical value*

*Social currency*

# Appendices

## Appendix A

# Analisi usabilità di un sito (\*)

Di seguito elenchiamo una serie di punti da seguire per svolgere un'attenta e meticolosa analisi di usabilità per un sito web. Qui si è provato a raccogliere una lista generale e quindi ovviamente alcune parti possono essere saltate o espanse, molto sta alla natura del sito. I punti di seguito sono estratti soprattutto dai capitoli teorici descritti in questo documento. I punti segnalati con l'asterisco (\*) non comprendono l'usabilità.

### A.1 Sito: note generiche

#### Nome dominio \*

Un sito che non attrae e presenta problemi già sul nome difficile abbia successo tra gli utenti.

#### SEO \*

Un sito può essere il master in usabilità ma se non ha un buon posizionamento SERP non serve a nulla.

- Keyword
- Posizionamento SERP
- Contatto con i social (pagerank 3.0)

#### Navigazione

- Breadcrump
- Richieste di registrazioni
- Design (bloated design)
- Menu di navigazione

## **Ricerca**

- Ricerca interna
- Modalità di ricerca disponibili
- Caratteristiche search box

## **Sito commerciale**

- Illustrazione del prezzo
- Illustrazione dei prodotti
- La pubblicità dei prodotti

## **Pubblicità**

- Posizione nella pagina
- Modalità in cui è comunicata

## **Mobile**

- Tempo caricamento
- Taglia schermo
- Interazione

## **A.2 Pagine: dettagli**

### **Pagina generica**

- Timer
- Testo
- Struttura del contenuto
- Scroll
- Comunicazione dei 6 assi
- Eventuali metafore visive

### **Homepage**

### **Error 404**

