

INDICE

1. Introduzione
2. Social Network Analysis
 - 2.1. Definizione di social network
 - 2.2. Storia della social network analysis
 - 2.3. Ultimi sviluppi
 - 2.4. Dati relazionali e loro rappresentazione
 - 2.4.1. Grafi per la rappresentazione dei social network
 - 2.5. Misure
3. Web 2.0 e WebApplications
 - 3.1. Vantaggi della piattaforma web
 - 3.2. AJAX
 - 3.3. Model View Controller
 - 3.4. Browser e tecnologie lato client
 - 3.4.1. HTML
 - 3.4.2. Javascript
 - 3.4.3. Document Object Model
 - 3.4.4. Dojo
 - 3.4.4.1. Architettura
 - 3.4.4.2. Onload() e Dojo.addonload()
 - 3.4.4.3. Accesso a DOM
 - 3.4.4.4. Eventi
 - 3.4.4.5. JSON
 - 3.4.4.6. Comunicazione client-server
 - 3.4.5. CSS
 - 3.4.6. Grafica Web
 - 3.4.6.1. SVG
 - 3.4.6.2. VML
 - 3.4.6.3. HTML canvas
 - 3.5. Programmazione lato server
 - 3.5.1. Motore di Database / Mysql
 - 3.5.2. HTTP server (Apache)
 - 3.5.2.1. Moduli
 - 3.5.3. Python
 - 3.5.3.1. Il linguaggio
 - 3.5.3.2. Tipi di dati e strutture
 - 3.5.3.3. Libreria standard
 - 3.5.3.4. Python per il web: mod_python
 - 3.5.3.5. Numpy e scipy
 - 3.5.4. Comunicazione client/server
 - 3.5.4.1. Json

- 3.5.4.2. Comunicazione tramite Javascript (synchronous/unsynchronous) e Dojo
- 4. Fase di Design
 - 4.1. Motivazioni
 - 4.2. Requisiti di base
 - 4.3. Le relazioni in azienda
- 5. Fase di Sviluppo
 - 5.1. Versione alpha
 - 5.2. Versione beta
 - 5.2.1. I moduli
 - 5.2.1.1. Modulo *CREATE*
 - 5.2.1.2. Modulo *SETUP*
 - 5.2.1.3. Modulo *DRAW CONVERSATION*
 - 5.2.1.4. Modulo *ANALYSIS*
 - 5.2.1.5. Network Analysis e raccolta dati
 - 5.2.1.6. Database
 - 5.2.2. Network Analysis e Raccolta dati
 - 5.2.3. Database
- 6. Test ergonomici
 - 6.1. Ergonomia
 - 6.1.1. Test di usabilità e Checklist
 - 6.1.2. L'analisi dei movimenti oculari
 - 6.2. Scopo della sperimentazione
 - 6.3. L'esperienza
 - 6.3.1. Le modifiche al software
 - 6.3.2. Soggetti
 - 6.3.3. Risultati
 - 6.3.3.1. Analisi Checklist
 - 6.3.3.2. Analisi del Questionario finale sull'Efficacia degli Algoritmi
 - 6.3.3.3. Analisi dei Task
 - 6.3.3.4. Analisi dei Blink
 - 6.3.3.5. Analisi dei Movimenti Oculari
- 7. Conclusioni
 - 7.1. Website Analysis
 - 7.2. Nuove Funzionalità Per l'Applicazione Esistente
- 8. Appendice
- 9. Bibliografia

1. Introduzione

Con l'avvento di internet e del Web 2.0 lo studio delle reti sociali, o Social Networks, ha acquistato sempre più importanza sociale, economica e politica. Grazie a siti come Facebook l'espressione "social network" è entrata nel vocabolario comune, ad indicare quei siti web, nei quale si possono contattare amici e conoscenti attraverso computer o dispositivi mobili (quali smartphone e tabletPC), facendoli partecipi dei propri pensieri, in ogni momento e in ogni luogo nel quale sia presente una connessione ad internet.

Il significato originale del termine si riferisce invece al concetto di *"un qualsiasi gruppo di persone connesse tra loro da diversi legami sociali, che vanno dalla conoscenza casuale, ai rapporti di lavoro, ai vincoli familiari"*. Molteplici sono infatti le pubblicazioni riguardanti l'argomento sin dagli anni '50 riguardanti vari aspetti sia matematico-scientifici, che ingegneristici, che psicologici, che economico-sociologici. Famosi a questo proposito sono le pubblicazioni del matematico ungherese Paul Erdős, e la teoria sociologica dei sei gradi di separazione, la quale afferma che *"qualunque persona può essere collegata a qualunque altra persona attraverso una catena di conoscenze con non più di 5 intermediari"*.

Lo scopo di questa tesi è quello di sviluppare un software per l'analisi delle reti sociali che si formano tra gruppi partecipanti a riunioni di vario tipo: dibattiti, consigli direttivi aziendali, focus groups, lezioni, assemblee, ecc.

Il software potrà essere utilizzato per la simulazione di gruppi di lavoro e per l'analisi di situazioni reali; il grafo della rete che si forma tra le persone intervenute verrà rappresentato al suo interno tramite icone o fotografie dei soggetti e archi che si andranno a tracciare all'atto di comunicazione tra due nodi/partecipanti. Sarà inoltre possibile calcolare, sia in tempo reale che durante un'analisi a posteriori, alcune statistiche utili ad identificare il livello di partecipazione dei soggetti alla discussione. Per l'uso in real-time durante la riunione saranno studiate le possibilità fornite dall'utilizzo di Notebook "Tablet" con touchscreen o pennino.

Il primo capitolo comprende una introduzione alle reti sociali e alla teoria dei Grafi applicata allo studio dei Social Network. Verranno presentati dei lavori correlati riguardanti l'utilizzo dei social network e la visualizzazione delle statistiche sulle reti, in alcuni esempi significativi.

Nel capitolo successivo, verrà descritta la tecnologia utilizzata per la realizzazione del software. Verrà spiegato in dettaglio l'utilizzo di Linguaggi client-side e server-side, il ruolo del web server e del DBMS per lo sviluppo di applicazioni web.

Nel terzo capitolo, verrà introdotto il software e la descrizione dettagliata delle sue funzioni e dei requisiti richiesti.

Nel quarto capitolo verrà mostrata l'implementazione del software dalla versione preliminare a quella stabile da utilizzare per i test.

Il quinto capitolo riguarderà i test ergonomici per il miglioramento della User Experience e della facilità di utilizzo.

Nell'ultimo capitolo verranno, infine, descritti alcuni possibili sviluppi in cui il lavoro potrà svilupparsi.

2. Social Network Analysis

2.1. Definizione di social network

Ogni tipo di struttura sociale può essere rappresentata con l'ausilio di un grafo, nel quale i nodi rappresentano gli attori sociali e gli archi le relazioni che intercorrono tra questi. Questo tipo di rappresentazione di è chiamata rete sociale, o *Social Network*. Ogni unità partecipante è, solitamente, chiamata *attore* (una persona, un'organizzazione, una nazione, un blog e così via).. L'insieme di possibili relazioni che si possono considerare è molto vasto: vicinanza, amicizia, opinione su un'altra persona, i rapporti commerciali, connessioni fisiche o virtuali, la presenza in una pagina web di un link. Secondo questa prospettiva, il punto di vista per l'analisi dei fenomeni sociali si sposta dagli individui e le loro caratteristiche alle relazioni che tra gli individui si formano permettendo, così, l'analisi processi sociali che avvengono all'interno della rete.

2.2. Storia della Social Network Analysis

Secondo Scott (1991) l'origine della Social Network Analysis è da attribuire agli studi effettuati da Jacob Moreno (1889-1974) sulla "sociometria", la scienza che studia i metodi di rilevazione e misurazione delle relazioni intercorrenti all'interno di un gruppo o comunità, e da Fritz Heider (1896-1988) sulla "Triad Equilibrium Analysis". Queste prime idee vennero successivamente riprese da Frank Harary (1921-2005) a Dorwin Cartwright (-1988) i quali individuarono nei metodi propri della teoria dei grafi un potente metodo formale per lo studio delle strutture sociali.

Intorno agli anni '30 e '40 alcuni ricercatori di Harvard, tra cui Lloyd Warner (1898-1970) e Eltan Mayo (1880-1949) in uno studio sulle comunità del New England e un successivo sulle condizioni di lavoro degli operai della "Chicago Central Plant", introdussero l'uso dei sociogrammi, una metodologia di indagine, usata ancora oggi nelle scienze dell'educazione, che tramite l'uso di questionari si pone l'obiettivo di ricostruire le posizioni degli individui all'internodi un gruppo, per la valutazione delle proprietà informali dei grafi definendo, inoltre, il concetto di "clique". Una profonda revisione critica di questi concetti venne operata da George Homans (1910-1989).

Un fondamentale sviluppo alla Social Network Analysis venne dato da due ricercatori del dipartimento di antropologia sociale della Manchester University, John Barnes e (1918-) e Siegfried Nadel (1903-1956), i quali puntarono l'attenzione sulle relazione date da potere e conflitti tra gli individui, piuttosto che dalle configurazioni sociali preimposte, quali rapporti gerarchici, focalizzandosi sull'analisi della struttura in luogo dell'analisi sul contenuto, mostrando come le relazioni visibili in una rete sociale siano molto diverse da quelle reali. Introdussero inoltre l'utilizzo di strumenti algebrici per l'analisi dei ruoli. Gli studi furono successivamente portati avanti da Clyde Mitchell (1918-1995) il quale introdusse la differenza tra rete "completa" e rete ego-centrica. Clyde introdusse, inoltre, l'utilizzo degli indici per l'analisi. Successivi studi di Harrison C. White (1930-) sostennero che la ricerca di proprietà del network non andava fatta basandosi su categorie prestabilite ma sulle reali proprietà della rete e, in questo contesto, Mark Granovetter dimostrò, inoltre, l'importanza dei legami deboli all'interno dei network.

Tra i vari tipi di analisi, una teoria originale, conosciuta come “teoria del mondo piccolo” e proposta da Stanley Milgram (1933-1984), si è rivelata essere piena di interessanti spunti di ricerca: essa afferma che in una rete sociale anche grande il percorso fatto di relazioni più breve necessario per collegare tra di loro due individui è solitamente molto piccolo rispetto alla dimensione della rete. Questa teoria, nonostante le molte critiche relative al paradigma di sperimentazione originale, si è rivelata un importante ambito di studio: le proprietà di questi “mondi piccoli” e la definizione di sistemi di questo tipo risultano ancora di notevole interesse scientifico. Approcci empirici al problema si sono rivolti anche al mondo web e ai network virtuali dando vita a sistemi quali Friendster e ai software di Instant Messaging.

Ai giorni nostri l'analisi dei Social Network interessa i più disparati campi di ricerca sociali, psicologici, antropologici, matematici, fisici, economici, e matematici.

2.3. Ultimi sviluppi

Tra i recenti sviluppi della ricerca sulla Social Network Analysis, una parte di studi ha cercato di trovarne delle applicazioni concrete in termini di strumenti di diagnosi dei gruppi sociale, di informazione e monitoraggio della rete in tempo reale. Verranno qui di seguito presentati alcuni esempi di approcci che utilizzano la Social Network Analysis per lo studio dei metodi di feedback per il monitoraggio delle relazioni sociali e per l'intervento su di esse rivolto al loro miglioramento della qualità delle connessioni tra i componenti, utilizzati come base per lo sviluppo del lavoro di questa tesi. Moltissimi altri esempi sono reperibili in letteratura.

Il primo caso si occupa invece del monitoraggio della salute sociale di un gruppo di persone anziane tramite l'ausilio di sensori e di sistemi di visualizzazione (Margaret E. Morris, 2005). Il network considerato in questo lavoro è quello che ha come soggetti un anziano e le persone che conosce e vede regolarmente. Questo tipo di network durante la vita di una persona è soggetto a grandi variazioni, e tende a diventare molto ristretto con l'aumento dell'età, rendendo gli anziani rassegnati alla solitudine. Inoltre, i contatti che un anziano ha con altre persone sono, molto spesso, mediati e favoriti da pochissime persone, come ad esempio figli ormai adulti, che diventano, perciò, nodi focali della rete che spesso sentono il peso di questa responsabilità. Per il monitoraggio delle relazioni sono stati utilizzati dei sensori che hanno permesso di monitorare parametri quali le chiamate telefoniche, la presenza di visite in casa e la loro durata, ai quali per completezza sono stati aggiunti dei report fatti dai vari soggetti, tramite intervista. I dati così raccolti sono stati resi disponibili ai vari attori sociali tramite alcune schermate di facile interpretazione a cui poteva avere accesso sia la persona anziana, sia un'altra persona che si prendeva cura dell'anziano (ad esempio, il figlio). Lo studio ha messo in evidenza come una conoscenza dello stato delle relazioni da parte dell'anziano ne migliori notevolmente la vita sociale e al contempo alleggerisca il carico di responsabilità dei nodi focali della rete, che prima erano completamente responsabili e con l'introduzione di questo feedback sentono venir meno il peso delle responsabilità.

Il secondo lavoro riguarda lo studio di un social network per un gioco on-line (Baù, Gamberini, Martino, Spagnoli, 2009). Scopo della ricerca era lo studio di come un particolare feedback sulla partecipazione al network dei soggetti influenzi l'attività del soggetto stesso. Lo studio era volto a comprendere quali tipi di feedback influenzavano maggiormente le attività comunicative entro una rete per un compito specifico. L'esperimento effettuato consisteva nell'utilizzo di un gioco on-line

chiamato “Crossfire” abbinato ad un client per la messaggistica istantanea per lo scambio dei messaggi tra i partecipanti. Il task dato ai soggetti era quello di ricercare, all’interno di una mappa fornita dall’ambiente di gioco, dentro la quale l’avatar del partecipante poteva muoversi liberamente, il maggior numero di calici nascosti, le quali informazioni per il ritrovamento erano fornite tramite dei cartelli situati in differenti città costitutive dell’ambiente. Gli utenti, divisi in gruppi, dovevano trovare il maggior numero possibile di oggetti per far vincere il match alla propria squadra. L’utente in ogni istante di tempo poteva vedere solo una determinata porzione di mappa: la ricerca poteva essere effettuata focalizzandosi sull’ambiente oppure utilizzando il gruppo per ricevere informazioni utili alla localizzazione dei calici. Sulla base dei dati raccolti manualmente sull’utilizzo della chat testuale e dei flussi di messaggi scambiati per comunicare con il resto del team vennero calcolati, in tempo reale, gli indici “Centrality Degree” e “Reciprocity” dei singoli soggetti (per una descrizione dettagliata degli indici si faccia riferimento al capitolo *misure*) dividendo i soggetti in tre gruppi (indipendenti dalla suddivisione in team), al primo fornendo come valore del feedback il valore di “Centrality Degree”, al secondo il valore di “Reciprocity” e al terzo nessun feedback. Per una facile interpretazione dei valori essi sono stati presentati all’utente tramite dei *gauge*. Infine ogni team doveva partecipare a tre match differenti per poterne studiare il trend.

Lo studio ha dimostrato che l’utilizzo di metodi feedback si dimostra efficace per il monitoraggio della rete e per l’incremento della comunicazione, benché questa maggior coesione sociale non sia rilevata dai soggetti esposti alla visione del feedback.

Il software sviluppato in questa tesi si propone di ideare delle modalità di raccolta dei dati programmatiche per migliorare la precisione dell’analisi delle reti. L’analisi di reti e di relazioni composti da persone, è ancora spesso, basata su raccolte di dati a campione o sulla base di analisi eseguite manualmente (come la lettura di questionari o le interviste). Le modalità di raccolta fornite dal software, descritte in dettaglio nella seconda parte di questo lavoro, si ritiene possano essere più precise e molto più simili alle metodologie utilizzate per la raccolta dei dati su reti digitali (come quelle web), e quindi molto utili per il miglioramento degli studi fatti su qualsiasi rete sociale.

2.4. Dati relazionali e loro rappresentazione

È importante definire che tipo di dati, all'interno di una rete sociale, sono adatti per essere studiati tramite i metodi dell'analisi di rete. Quando c'è la necessità di studiare un "social network", spesso i dati da raccogliere, utili per l'indagine da svolgere, non sono misurabili con l'ausilio di strumenti di misura scientifici, ma vanno raccolti con tecniche socio/psicologiche e sono soggetti, in parte, a interpretazione.

Sulla base di simili processi di interpretazione si possono distinguere tipi distinti di dati, cui corrispondono distinti metodi di analisi. I principali tipi di dati sono gli "attributi" e i dati "relazionali".

Gli *attributi* si riferiscono "agli atteggiamenti, alle opinioni e ai comportamenti di individui o gruppi, considerati come proprietà, qualità o caratteristiche appartenenti ai soggetti agenti". Le informazioni, raccolte solitamente attraverso sondaggi o interviste, vengono spesso considerate come attributi di singoli individui, suscettibili di essere quantificati e analizzati attraverso le tecniche statistiche disponibili. I metodi appropriati agli attributi sono quelli dell'*analisi delle variabili*, in cui gli attributi sono misurati come valori di particolari variabili.

I *dati relazionali*, invece, si riferiscono "ai contatti, ai vincoli e ai collegamenti, alle appartenenze e agli incontri di gruppo, che mettono in relazione un attore con l'altro e non possono quindi essere ridotti a proprietà degli stessi individui agenti". Le relazioni non sono proprietà degli attori, ma di sistemi di attori; esse collegano coppie di attori in sistemi relazionali. I metodi appropriati ai dati relazionali sono proprio quelli dell'analisi delle reti, in cui le relazioni sono esaminate in quanto esprimono i legami che intercorrono fra gli attori. Anche se è possibile, ovviamente, eseguire conteggi quantitativi e statistici delle relazioni, l'analisi delle reti comporta un corpo di misure qualitative della struttura del network.

I dati relazionali possono venire rappresentati in vari modi, ma non con le usuali matrici "casi-per-variabili". A esempio:

	Età	Sesso	Reddito	Ecc.
1				
2				
3				
4				
...				
Ecc.				

Infatti questo tipo di rappresentazione descrive dettagliatamente gli attributi dei singoli soggetti, ma non da alcuna informazione sulle relazioni presenti tra i soggetti.

La rappresentazione dei dati relazionali richiede un formato che metta in evidenza le relazioni tra i soggetti e non le loro proprietà. Una rappresentazione tabellare utile per la rappresentazione dei dati

relazionali è detta “*matrice di incidenza*”, nella quale si rappresentano le persone coinvolte nell’indagine nelle righe e gli eventi considerati nelle colonne. Quando uno specifico individuo prende parte ad un evento, nella corrispondente cella della matrice è posto un “1”, la non-partecipazione è indicata con uno “0”. Questo tipo di rappresentazione mostra quali soggetti hanno partecipato a quali eventi e ci indica perciò la relazione tra di loro: se due soggetti o più hanno partecipato ad un stesso evento allora saranno in relazione tra di loro. Da questo tipo di rappresentazione ne sono state poi ideati altri, molto più simili alle matrici che verranno utilizzate, poi, nell’analisi delle reti tramite la teoria dei grafi: le tabelle “caso-per-caso”. Questo tipo di rappresentazione, che pone su righe e colonne i soggetti, mostra direttamente le relazioni che intercorrono tra essi, mostrando nel campo corrispondente un “1” se i soggetti sono in relazione e uno “0” se, invece, non lo sono. Un altro tipo di rappresentazione possibile è detta “affiliazione-per-affiliazione” nella quale, in una tabella eventi x eventi vengono indicati gli eventi che hanno almeno un soggetto partecipante ad entrambi. Questi ultimi due tipi di rappresentazione sono detti *matrici di adiacenza*.

I dati contenuti nella matrice possono essere suddivisi secondo due importanti criteri: Direzionalità e Numerazione. Per direzionalità si intende il grado di simmetria di una relazione. Per numerazione si intende il tipo di dati contenuto nei campi: binario se il campo può contenere solo i valori “0” oppure “1”, numerico se il campo può contenere qualsiasi numero positivo. In questo caso il valore viene detto “*peso*”

Per comprendere questi è possibile fare alcuni esempi.

Si considerino 4 società di un gruppo di aziende, dette “Società 1” “Società 2” “Società 3” e “Società 4”, e 5 direttori (A, B, C, D ed E). Si supponga che questi direttori controlli più di una azienda ciascuno, e ciascuna azienda abbia un Consiglio di amministrazione formato da più di un direttore. È possibile rappresentare in una matrice di incidenza la situazione:

	A	B	C	D	E
1	1	1	1	1	0
2	1	1	1	0	1
3	0	1	1	1	0
4	0	0	1	0	1

La quale indica, per esempio, che il direttore A controlla la società 1 e la società 2, mentre la società 1 è controllata da A,B,C e D ma non da E. Da qui è possibile, inoltre ricavare le due matrici di adiacenza:

	1	2	3	4
1	-	3	3	1
2	3	-	2	2
3	3	2	-	1
4	1	2	1	-

Che mostra, ad esempio, che le società 1 e 2 sono controllate da 3 amministratori in comune

	A	B	C	D	E
A	-	2	2	1	1
B	2	-	3	2	1
C	2	3	-	2	2
D	1	2	2	-	0
E	1	1	2	0	-

Che indica, ad esempio, che gli amministratori A e B controllano due aziende in comune

Considerando le matrici di adiacenza si può notare una simmetria. Questo può essere verificato anche utilizzando il modello da cui i dati derivano, in quanto se la società x ha n direttori in comune con la società y, allora anche la società y avrà n direttori in comune con la società x. In questo caso si dice che i dati sono "non orientati" in quanto la relazione $a \rightarrow b$ è uguale alla relazione $b \rightarrow a$ (e le matrici di adiacenza sono simmetriche). I dati contenuti nelle celle sono numerici in quanto contengono valori diversi da "0" e "1".

Un secondo esempio è quello che sta alla base di questo lavoro. Si consideri una riunione alla quale sono presenti 5 persone sedute attorno ad un tavolo, e come relazione da studiare le conversazioni avvenute tra un partecipante e un altro (per semplicità, per il momento si considerino solo le conversazioni tra due utenti e non quelle tra 3 o più persone). Questo tipo di relazioni necessita di due ruoli ben distinti: una sorgente ed un ricevente del messaggio, una relazione perciò di natura asimmetrica.

In questo caso la matrice di incidenza è di difficile interpretazione, avendo a che fare con parecchi eventi (in figura è rappresentata la tabella con i soli primi 9 eventi):

	Conv. 1	Conv. 2	Conv. 3	Conv. 4	Conv. 5	Conv. 6	Conv. 7	Conv. 8	Conv. 9
A	1	0	0	1	1	0	1	0	1
B	1	1	0	1	0	1	0	1	1
C	0	1	1	0	0	0	1	0	0
D	0	0	0	0	1	0	0	1	0
E	0	0	1	0	0	1	0	0	0

Molto più utile è la matrice di adiacenza che mette invece in relazione quante conversazioni sono state effettuate tra ogni singola coppia di utenti:

	A	B	C	D	E
A	-	3	54	6	11
B	4	-	54	19	32
C	5	5	-	59	45
D	1	8	22	-	19
E	32	13	7	34	-

Infatti, in questo tipo di matrice si può facilmente notare come, ad esempio l'utente E abbia parlato molto all'utente C ($E \rightarrow C = 45$), ma abbia ricevuto da questo poche risposte ($C \rightarrow E = 7$). Eventi di questo tipo sono detti non orientati.

2.4.1. Grafi per la rappresentazione dei social network

Una base comune per i programmi dell'analisi delle reti sociali è l'approccio matematico della teoria dei grafi, che fornisce un linguaggio formale per la descrizione delle reti e dei loro caratteri.

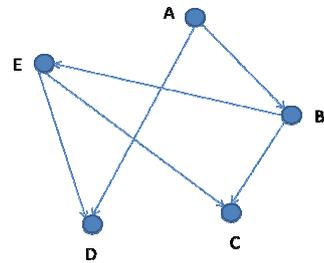
“Si definisce *grafo non orientato* una coppia $G=(V,E)$ in cui V è un insieme finito di elementi detti *nodi* o *vertici* ed E è una famiglia di coppie non ordinate di elementi appartenenti a V . Agli elementi di E , detti *archi* o *lati*, è possibile associare una funzione $f:E \rightarrow \mathbb{R}$, in questo caso gli archi vengono detti *archi pesati*. Se $\{i,j\} \in E$ compare 2 o più volte all'interno dell'insieme si parla di archi *multiplici*.”

Si definisce *grafo orientato* una coppia $G=(V,A)$, in cui V è un insieme finito di elementi detti *nodi* o *vertici*, e A è un insieme di coppie ordinate di vertici.” [rif]

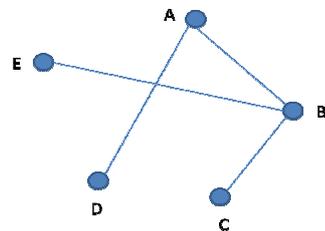
È possibile rappresentare i grafi tramite due soluzioni di facile interpretazione: la matrice di adiacenza (presentata nel paragrafo precedente), e il diagramma archi-nodi nel quale sono

rappresentati i nodi, numerati, collegati da archi. Come esempio è possibile rappresentare le matrici di adiacenza qui sotto tramite i grafi in figura.

	A	B	C	D	E
A	-	1	0	1	0
B	0	-	1	0	1
C	0	0	-	0	0
D	0	0	0	-	0
E	0	0	1	1	-



	A	B	C	D	E
A	-	1	0	2	0
B	-	-	1	0	3
C	-	-	-	0	0
D	-	-	-	-	0
E	-	-	-	-	-



per convertire la rappresentazione matriciale in rappresentazione grafica è possibile considerare le etichette delle colonne della matrice di adiacenza come elementi dell'insieme dei nodi. È necessario poi aggiungere gli archi a seconda dei casi: nel caso di un grafo orientato è necessario aggiungere un arco per ogni cella che contiene un valore maggiore di 0; Nel caso di un grafo non orientato è invece necessario utilizzare la funzione $v(x,y)+v(y,x)$, con $v(x,y)$ il valore contenuto nella cella (x,y) della tabella.

Nel caso sia importante considerare il peso dell'arco è possibile aggiungere un'etichetta ad ogni arco riportando il valore della funzione peso.



Per l'utilizzo dei grafi nella rappresentazione delle reti sociali si possono considerare i nodi in luogo dei soggetti presi in considerazione nello studio, e gli archi in luogo delle relazioni che si instaurano tra i soggetti. Se la relazione è asimmetrica il grafo risulterà orientato, altrimenti il grafo sarà non orientato. Se in una relazione sarà importante considerare la forza di legame che una relazione esprime tra due soggetti, il grafo generato sarà pesato.

In questo lavoro verranno analizzate reti sociali con relazioni asimmetriche di diverso peso. Il grafo generato sarà quindi pesato e orientato.

2.5. Misure

Oltre all'informazione grafica resa dalle varie rappresentazioni della rete, tramite l'utilizzo della teoria dei grafi è possibile utilizzare particolari indici, calcolati a partire dalla matrice di adiacenza, per l'analisi e l'interpretazione della rete e delle sue componenti. Gli indici utilizzabili si dividono in due categorie.

Si parla di indici "globali" quando l'informazione che si può ricavare da essi si riferisce alla struttura macroscopica della rete, come ad esempio il grado di densità delle relazioni oppure la loro omogeneità rispetto a tutti i nodi.

Gli indici vengono invece detti "locali" quando esprimono informazioni relative ai singoli nodi. Questi indici, solitamente, utilizzano algoritmi che necessitano in input dati specifici riguardanti il soggetto, come ad esempio il numero di relazioni uscenti dal determinato nodo, oltre ai dati globali relativi alla topologia della rete. Le informazioni ricavate da questo tipo di indici possono riguardare, ad esempio, il grado di importanza di un nodo all'interno della rete oppure la sua attività all'interno della discussione (nel caso delle reti sociali considerate in questo lavoro).

A seconda della tipologia di rete considerata nello studio sono stati ideati diversi indici, adatti a descrivere caratteristiche differenti.

Gli indici considerati per la tipologia di rete da analizzare sono:

- Densità (globale)
- centralizzazione (globale)
- reciprocità di gruppo (globale)

- grado (legato ai soggetti)
- grado in entrata (legato ai soggetti)
- grado in uscita (legato ai soggetti)
- reciprocità locale (legato ai soggetti)

La motivazione della scelta è stata la necessità di avere a disposizione valori significativi per piccoli social network. Si è cercato, inoltre, di utilizzare indici normalizzati, cioè aventi valori oscillanti tra 0 e 1, e calcolabili utilizzando algoritmi di complessità computazionale lineare.

2.5.1. Centralità

Il concetto di centralità sta ad indicare la persona più "popolare" nel proprio gruppo o quella che è al centro dell'attenzione. Il modo più semplice per misurare la centralità consiste nel far riferimento ai gradi dei vari nodi, cioè al numero di altri nodi cui il soggetto è direttamente collegato (tramite un arco). Questo tipo di misura è facilmente ricavabile dalla matrice di adiacenza. Il grado del nodo i sarà:

$$D_i = \sum_{j=1}^n \begin{cases} 0, & \text{se } v(n_i, n_j) + v(n_j, n_i) = 0 \\ 1 & \text{se } v(n_i, n_j) + v(n_j, n_i) \neq 0 \end{cases}$$

Con n il numero massimo di nodi e $v(a,b)$ il valore contenuto nella cella (a,b) della matrice.

Nel caso di grafi pesati il valore di centralità, benchè ancora calcolabile utilizzando le stesse formule, tende a presentare una bassa significatività, specie con reti piccole, in quanto i vari nodi del grafo sono spesso tutti collegati tra loro con un numero di connessioni molto elevato. Queste connessioni presentano però pesi molto differenti.

Si consideri ad esempio il caso della riunione: all'entrata dei partecipanti probabilmente avverranno i saluti e le presentazioni. Verranno perciò registrati molti eventi di comunicazione tra i partecipanti, che andranno a riempire l'insieme degli archi/relazioni della rete portando a valori molto vicini al massimo il grado di tutti i soggetti. Durante lo svolgimento vero e proprio della riunione è poi probabile che i partecipanti parleranno di più con le persone sedute vicine oppure con il coordinatore della riunione, piuttosto che con quelle sedute lontane. Con il proseguimento della riunione si registrano pesi molto differenziati tra le varie relazioni instaurate. In questi casi si otterranno perciò grafi completi ma pesati, con pesi molto differenti da arco ad arco. Per utilizzare questo dato e aumentare la significatività della misura è possibile utilizzare la dicotomizzazione della matrice (spiegata alla fine di questo capitolo)

Per normalizzare i risultati, il grado viene inoltre diviso per il numero di nodi presenti nella rete:

$$\bar{D}_i = \frac{D_i}{n}$$

In un grafo orientato possono, inoltre, venire utilizzati altri due indici per la misura della centralità di un soggetto all'interno della rete: In-Degree e Out-Degree.

Con In-Degree si intende il grado del soggetto considerando, però solo gli archi/relazioni diretti verso di esso:

$$I_i = \sum_{j=1}^n \begin{cases} 0, & \text{se } v(n_j, n_i) = 0 \\ 1 & \text{se } v(n_j, n_i) \neq 0 \end{cases}$$

Con Out-Degree si intende, invece, il grado del soggetto considerando solamente gli archi/relazioni che dal soggetto si dirigono verso altri:

$$O_i = \sum_{j=1}^n \begin{cases} 0, & \text{se } v(n_i, n_j) = 0 \\ 1 & \text{se } v(n_i, n_j) \neq 0 \end{cases}$$

Anche in questi casi gli indici possono venire normalizzati:

$$\bar{O}_i = \frac{O_i}{n}$$

$$\bar{I}_i = \frac{I_i}{n}$$

2.5.2. Reciprocità

I vari indici di centralità sono utili, oltre che per valutare l'importanza di un certo soggetto all'interno di una rete, anche per valutare la sua attività all'interno. Infatti, un soggetto è molto attivo quando è molto in relazione con gli altri o quando gli altri sono in relazione con lui, e a seconda dei casi sarà attore o ascoltatore passivo. Questo tipo di misurazione, però, da questo punto di vista possiede delle limitazioni: non descrive la simmetria della rete, cioè non descrive se la tendenza di un soggetto è quella di rispondere alle richieste oppure quella di ascoltare o ancora quella di parlare senza ascoltare.

Come misura complementare a quelle di centralità può quindi essere usata quella di reciprocità, che esprime il rapporto tra il numero di scambi simmetrici rispetto al totale (calcolo da effettuare sulla matrice orientata e non dicotomizzata). Il codice per il calcolo (in Python) è riportato di seguito:

```
for i in range(length):
    row = snamatrix[i,:]           //estraggo le righe dalla matrice snamatrix
    column = snamatrix[:,i]       //estraggo le colonne dalla matrice
    for j in range(length):
        if row[j] > 0 or column[j] > 0:
            tot[i] += 1.
            if row[j] > 0 and column [j] > 0:
                sym[i] += 1.
    if tot[i] == 0:
        tot[i] = 1
return sym/tot
```

È possibile, inoltre, un ulteriore indice ad indicare il grado totale di reciprocità del grafo calcolato come la media delle reciprocità dei soggetti:

$$R_{group} = \frac{\sum_{i=1}^n R_i}{n}$$

2.5.3. Densità

Uno dei concetti più largamente usati nella teoria dei grafi è quello di densità, che descrive il livello generale dei legami fra i punti in un determinato grafo. Un grafo si dice "completo" quando tutti i punti sono adiacenti l'uno all'altro: ogni punto è collegato direttamente a ognuno degli altri, il concetto di densità è un tentativo di sintetizzare la distribuzione globale delle linee per misurare quanto il grafo sia lontano da queste situazioni di completezza. Quanto più numerose sono le linee direttamente collegate tra loro quanto più il grafo sarà denso.

La densità dipende da due fattori: l'inclusività del grafo e la somma dei gradi dei suoi punti.

L'inclusività si riferisce al numero dei punti che sono inclusi nelle varie parti collegate del grafo, cioè il numero totale di punti meno quello dei punti isolati (nodi del grafo che non sono

collegati a nessun altro). Ad esempio un grafo di 20 punti con cinque punti isolati avrà una inclusività pari a 0,75. Il concetto di inclusività e di densità sono collegati in quanto un nodo isolato non è incidente con nessuna linea e quindi non contribuisce alla densità del grafo. Di conseguenza tanto più un grafo è inclusivo, tanto più sarà anche denso. Di contro, però, il numero di connessioni tra i vari nodi può sempre variare: alcuni nodi avranno molti collegamenti con altri, mentre alcuni potrebbero averne connessioni meno numerose. Tanto più le connessioni saranno numero tanto più la densità sarà maggiore. Per misurare la densità si utilizza una formula che incorpora questi due parametri e confronta il numero effettivo di archi presenti nel grafo con il numero di archi che il teorico grafo completo con lo stesso numero di nodi avrebbe.

Il numero delle linee potenzialmente presenti nel grafo completo si può facilmente calcolare utilizzando la formula: $n(n-1)/2$ nel caso di un grafo non orientato. Nel caso di un grafo orientato il numero di possibili archi è doppio in quanto per ogni coppia di nodi x e y ci saranno due archi (x,y) e (y,x) . la formula da utilizzare sarà quindi: $n(n-1)/2*2=n(n-1)$. Il numero di archi presenti nel grafo considerato è invece calcolabile tramite la matrice di adiacenza del grafo, tramite gli algoritmo (per il grafo pesato):

```
def symmetrise(matrix):                                #procedura per la simmetrizzazione della
matrice                                               matrice
    length = sizeX(snamatrix)                          #contiene la dimensione della
matrice                                               matrice
    for i → 1 to length:
        for j → 1 to length:
            if (matrix[i, j] > 0) or (matrix[j, i] > 0):
                symm_matrix[i, j] = 1
                symm_matrix[j, i] = 1
    return symm_matrix

def density(matrix):                                  #in input la matrice
matrix                                               matrice
    length = sizeX(matrix)                              #contiene la dimensione della
matrice                                               matrice
    symm_matrix = symmetrise(matrix)
    return symm_matrix.sum()/(float(length)*float(length-1))

con symm_matrix.sum() somma degli elementi della matrice
```

Le formule per il calcolo della densità sono dunque:

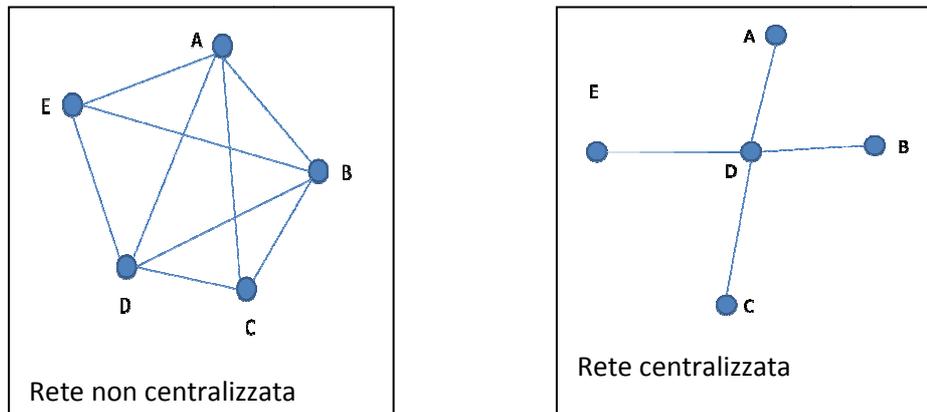
- Per il grafo non orientato $\frac{l}{n(n-1)/2}$
- Nel caso di grafo orientato $\frac{l}{n(n-1)}$

Dalle due definizioni si può ricavare che il valore dell'indice densità oscilla tra 0 (nel caso di nodi tutti isolati) e 1 (nel caso di un grafo completo).

Anche in questo caso, per migliorare la significatività dell'indice può essere utilizzata per il calcolo della densità in un grafo pesato, la matrice dicotomizzata.

2.5.4. Centralizzazione

I concetti di densità e centralizzazione, benché entrambi atti a descrivere la struttura della rete, si riferiscono ad aspetti differenti della “compattezza” del grafo. La densità descrive il livello generale di coesione. La centralizzazione descrive la misura in cui questa coesione è organizzata intorno a particolari punti focali. Centralizzazione e densità sono perciò importanti misure complementari.



Esistono varie procedure per il calcolo della centralizzazione, basate sul calcolo del rapporto tra la somma reale delle differenze di centralità e la loro somma massima possibile.

Per la procedura è utilizzato la matrice di adiacenza dicotomizzata del grafo orientato. La formula utilizzata in questo lavoro è:

$$C = \frac{\sum_{j=1}^g C_a(n^*) - C_a(n_j)}{(n-1)(n-2)}$$

Dove al numeratore è presente la sommatoria delle differenze tra il valore di centralità massimo $C_a(n^*)$ e quello dei singoli nodi (n_j). Al denominatore è presente invece il massimo valore teorico raggiungibile dal numeratore. Questo valore corrisponde a $(n-1)(n-2)$ con n numero di nodi.

2.5.5. Procedura di dicotomizzazione

Per migliorare la significatività di alcuni indici, come ad esempio la densità (che presenta le problematiche descritte precedentemente) per il calcolo viene utilizzata la tecnica della dicotomizzazione della matrice di adiacenza. La matrice di adiacenza viene “preprocessata” in modo da sostituire ai vari pesi il valore 0 oppure 1 a seconda che all’interno della cella considerata corrispondente alla relazione sia presente un valore maggiore o minore del valore, detto *cutpoint*, calcolato partendo dai vari pesi.

L'algoritmo per il calcolo del cutpoint è (codice in python, tramite l'utilizzo della libreria Numpy):

```
matrix = //matrice dei dati di adiacenza
temp = [] #matrice che contiene tutti i
          valori diversi da 0 della
          matrice

for i in range(size(matrix, axis = 0)):
    for j in range(size(matrix, axis = 0)):
        if (i != j) and matrix[i,j]!=0:
            temp.append(matrix[i,j])
arrTemp = array(temp)
return numpy.mean(temp)
```

con `numpy.mean(temp)` la media dei valori contenuti nella matrice

3. Web 2.0 e WebApplications

La tecnologia scelta per lo sviluppo dell'applicazione consiste nella creazione di un sito web dinamico. Per sito web dinamico si intende un sito che, utilizzando linguaggi come javascript e PHP, permette all'utente di interagire con la pagina HTML, così come si fa con una normale applicazione compilata in codice nativo o pseudocodice (ed es. Java). L'elaborazione grafica avviene nel client, mentre il mantenimento dei dati, il loro reperimento e le elaborazioni più pesanti computazionalmente sono effettuate nel server, che andrà perciò dimensionato di conseguenza dal punto di vista della memoria e della potenza di calcolo. L'utilizzo di tecniche cosiddette AJAX permette inoltre l'utilizzo di una stessa applicazione su qualsiasi piattaforma Hardware-Software fornita di un moderno browser per la navigazione in internet.[4]

Con il termine *web application*, si intendono tutte quelle applicazioni alle quali si accede tramite una rete, intranet o Internet che sia. Il termine inoltre indica un software che è ospitato in un ambiente browser o scritto in un linguaggio supportato da browser, il quale è necessario per rendere l'applicazione eseguibile. (In questo lavoro il termine verrà utilizzato in questa seconda accezione.)

3.1. Vantaggi della piattaforma web

L'utilizzo di una piattaforma web-based per lo sviluppo e la successiva distribuzione dell'applicazione presenta alcuni vantaggi ma introduce anche alcune problematiche.

Il primo vantaggio è la possibilità di accedere al software e alle informazioni presenti nel server virtualmente in qualsiasi luogo: le informazioni e i file creati sono infatti conservati in un unico database centralizzato e accessibile tramite la rete. Questo fatto introduce però anche alcune problematiche di sicurezza: le informazioni sono accessibili a tutti gli utenti della rete, e quindi è necessario sviluppare un solido sistema di autenticazione per nascondere le informazioni sensibili. Inoltre il responsabile per la conservazione dei dati diventa il fornitore dell'applicazione che dovrà tutelare i dati con opportuni backup del sistema.

Un secondo vantaggio è la facilità di installazione, distribuzione, e aggiornamento, poiché tutte queste operazioni vengono svolte una volta sola esclusivamente sul server che ospita l'applicazione. All'atto dell'aggiornamento tutti i terminali che utilizzano l'applicazione saranno automaticamente. Il supporto risulta essere molto più semplice ed economico.

3.2. AJAX

Asynchronous JavaScript and XML, meglio conosciuto con la sigla AJAX, non è un linguaggio di programmazione vero e proprio, bensì una collezione di tecniche multi-piattaforma utilizzabili su molti sistemi operativi, architetture informatiche e browser web, della quale esistono numerose implementazioni open source in librerie e framework.

La tecnica Ajax consiste in:

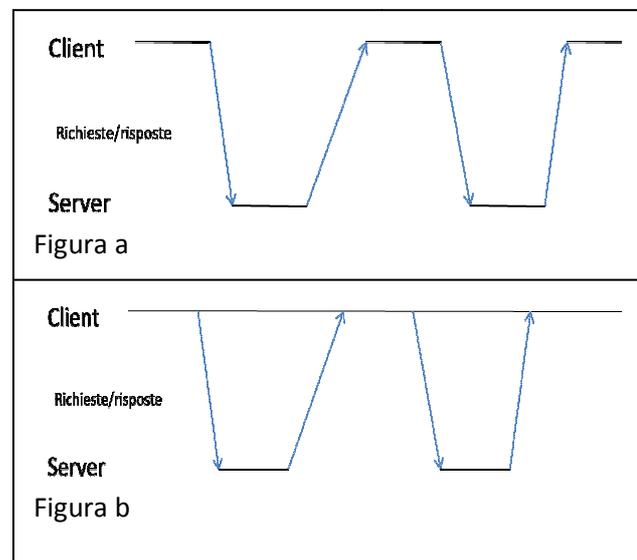
- *HTML* (o *XHTML*) e *CSS*: per la definizione della pagina web e dello stile;

- **DOM (Document Object Model):** manipolato attraverso linguaggi di script come JavaScript per l'elaborazione lato client e la gestione della Graphical User Interface
- **l'oggetto XMLHttpRequest:** componente dei moderni browser, accessibile tramite javascript, per l'interscambio asincrono dei dati tra il browser dell'utente e il web server. Questo permette la modifica grafica della sola porzione di pagina web utilizzata senza la necessità di effettuare un nuovo caricamento completo della pagina come avviene per le pagine html "statiche", ottenendo una fluidità maggiore dell'applicazione web, e un'esperienza utente più simile a quelle delle abituali applicazioni desktop. È grazie a questa caratteristica che la metodologia AJAX si è affermata per la distribuzione di applicazioni web.
- **XML:** formato di scambio dei dati, anche se di fatto qualunque formato può essere utilizzato, incluso testo semplice, HTML preformattato, JSON e perfino EBML. Questi file sono solitamente generati dinamicamente da script lato server e client. In questo lavoro il formato utilizzato per l'interscambio dati tra client e server è **JSON**

Le applicazioni web che usano Ajax richiedono browser che supportino le tecnologie utilizzate nell'implementazione della tecnica. Questi browser possono includere: Mozilla Firefox, Opera, KDE Konqueror, Apple Safari, Google Chrome e Microsoft Internet Explorer.

La chiave per comprendere l'impatto che Ajax ha avuto nello sviluppo web si trova nel miglioramento dello "user workflow". Con *Workflow* si intende il modo in cui l'utente interagisce con l'applicazione. Una buona applicazione rende l'utente produttivo ricreando il suo schema di lavoro, perciò un'applicazione che rallenta l'utente a causa delle sue limitazioni tecniche riduce la produttività.

In figura è mostrato il workflow pre-AJAX. In questo modo di esecuzione l'applicazione presenta informazioni all'utente oppure attende la risposta del server. Dal punto di vista dell'utente questo è molto frustrante, in quanto esso è costretto ad aspettare la risposta del server senza poter interagire con l'applicazione. In figura 2 è invece mostrato come AJAX modifica il workflow. Infatti grazie all'oggetto XMLHttpRequest e alle chiamate asincrone, l'interfaccia utente rimane attiva mentre il server è occupato eliminando le continue distrazioni al flusso di lavoro dell'utente.



La metodologia AJAX risente però di alcuni problemi:

- Molto spesso le applicazioni scritte in AJAX sono costituite da una sola pagina, rendendo inutilizzabile e a volte anche inutile l'uso del pulsante "Back", presente in tutti i browser e utilizzato per la navigazione a ritroso tra le pagine web precedentemente scaricate dall'utente finale. Se si ritiene che questa funzionalità sia necessaria, molto spesso sarà necessario creare un proprio sistema di "Undo" personalizzato.

- Il refresh risulta problematico, in quanto lo stato client-server potrebbe risultare desincronizzato, compromettendo la stabilità del server o l'allineamento dei dati. Un altro grosso problema è la compatibilità con i vari browser: infatti i moderni browser in commercio non supportano pienamente i vari standard, rendendo perciò necessario delle implementazioni browser-specifiche di alcune parti del codice, rendendo complicato il mantenimento del software stesso. Problemi di questo tipo si hanno inoltre dovendo supportare varie versioni di uno stesso browser, infatti l'evoluzione delle funzionalità e delle specifiche supportate è molto variabile tra un aggiornamento e il successivo.
- È necessario, poi, adattare l'applicazione per la visualizzazione della pagina a varie risoluzioni: essendo visualizzabile da dispositivi molto differenti (notebook widescreen e non, cellulari, smartphone, palmari ecc.) questa caratteristica può diventare problematica, rendendo necessario sviluppare parti di codice e di interfaccia grafica adattate ad ogni dispositivo utilizzato. Infine anche la gestione degli eventi va realizzata tenendo conto delle varie modalità di interazione possibili: mouse, touchscreen, pulsantiera, ecc. A questo proposito i vari browser utilizzati forniscono un supporto differente agli eventi, legato al tipo dei dispositivi per i quali sono stati realizzati.

Per ovviare a questi problemi mantenendo, nel contempo, le caratteristiche innovative della metodologia sono state introdotte varie soluzioni e framework. Per questa lavoro è stato utilizzato il framework Dojo. Tra le varie soluzioni esistenti, si è scelto questo framework, per il tipo di licenza open Source con cui viene distribuito, e per l'enorme quantità di documentazione presente in rete.

3.3. Model View Controller

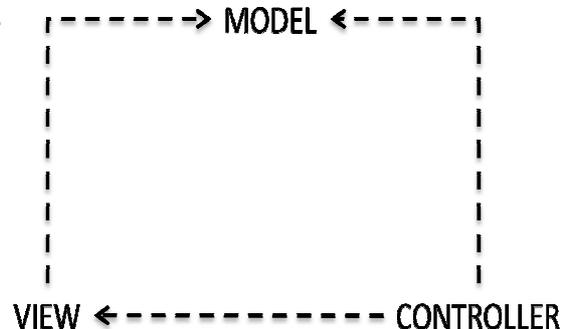
Come descritto precedentemente, la tecnologia web si avvale di un *browser* per l'elaborazione di una pagina web presente in un server centrale collegato ad Internet o ad una intranet il quale si occupa di fornire la pagina (o le pagine) costitutive dell'applicazione, di reperire i dati su richiesta dell'utente e di mantenerli, e di effettuare i calcoli più pesanti sui dati. L'elaborazione della grafica e il mantenimento di una parte dei dati temporanei avverrà nel client tramite le funzionalità fornite dai moderni browser.

Il paradigma utilizzato per la programmazione di applicazioni Web viene detto *Model-View-Controller* (o *MVC*) e separa in tre componenti il software da sviluppare:

- *Model*: Questo componente gestisce il comportamento dell'applicazione e l'acquisizione dei dati, gestisce gli stati di funzionamento, e risponde alle istruzioni per il cambiamento dello stato
- *View*: Questo secondo componente si occupa della visualizzazione dei dati e dell'interfaccia grafica
- *Controller*: Il controller interpreta gli eventi provenienti da mouse, tastiere e altri dispositivi di input e si occupa di avvisare Model e View di apportare gli opportuni cambiamenti

In figura è mostrato lo schema generale di come questi componenti interagiscono tra loro. È

importante notare come entrambi i componenti View e Controller dipendano dal Model, e quest'ultimo non dipenda da nessuno dei due. Questo è l'importanza centrale della separazione. Questa separazione permette al Model (che sostiene alle logiche di base dell'applicazione) di essere scritto e testato indipendentemente da come i dati verranno poi presentati e da come l'applicazione risponderà alle



richieste dell'utente. L'indipendenza di View da controller permette, inoltre, di sviluppare un'interfaccia grafica quasi completamente indipendente dal tipo di interazione effettuata su di essa dal particolare dispositivo di input che l'utente utilizzerà.

Grazie alle proprietà di separazione introdotte, il testing dell'applicazione risulta molto semplificato, infatti è possibile testare indipendentemente le tre logiche. Si semplifica inoltre il problema del testing delle Interfacce grafiche. Solitamente il test delle UI richiede un test manuale oppure la creazione di script che simulano il comportamento dell'utente. MVC non elimina la necessità di testare tali interfacce ma grazie alla separazione tra dati, loro presentazione e gestione degli eventi, semplifica tale verifica riducendo il numero di test da effettuare.

Sempre riguardo le interfacce MVC, poiché in esso View è separata dal resto, e non ci sono dirette dipendenze di View dal Controller, è possibile progettare diverse viste simultanee per la visualizzazione dei dati: ad esempio, varie pagine in una applicazione web possono accedere agli stessi dati e visualizzarli in maniera differente; è anche possibile progettare diversi tipi di visualizzazione dipendenti dalla risoluzione dello schermo e dalle caratteristiche del dispositivo. Inoltre l'interfaccia solitamente tende a cambiare più velocemente della "Business Logic", cioè dello

strato applicativo adibito alla gestione e al mantenimento dei dati , e grazie alla separazione del Model da View, le modifiche risultano più veloci.

MVC introduce però un nuovo livello logico, non bidirezionale, e questo può provocare un aumento di complessità al momento della progettazione dell'applicazione. Infatti se il Model necessita di aggiornamenti frequenti, necessariamente sia View che Controller dovranno essere modificati di conseguenza. È quindi necessario aver bene in mente il modello di visualizzazione fin dal momento della progettazione.

Esistono alcune varianti a MVC, la più comune fra tutte è la fusione tra View e Controller (modello Document-View) che presenta il vantaggio di accoppiare l'interfaccia con la gestione degli eventi per una migliore interazione fra le due componenti.

Nelle moderne applicazioni web il lato client implementa View e Controller del modello MVC. Il componente Model è invece spezzato tra client e server.

Verrà ora descritta in dettaglio la tecnologia utilizzata lato client e lato server e come il paradigma MVC può essere implementato in una applicazione web.

Il lato client si servirà di un browser come piattaforma per l'elaborazione del codice HTML e javascript costitutivo della pagina, e per la visualizzazione grafica. Lato server verrà, invece, utilizzata la tecnologia del web server dinamico e del linguaggio python. Si utilizzerà, inoltre, un server di database (DataBase Management System) per la gestione dei dati.

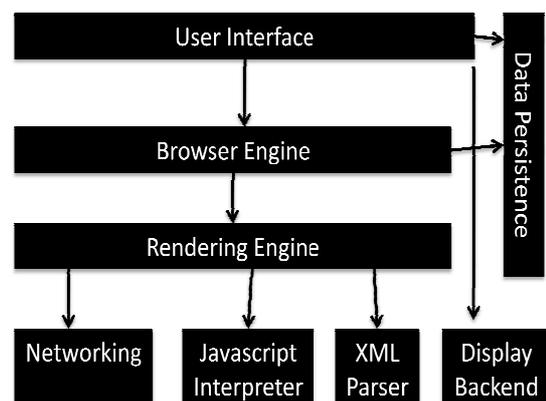
3.4. Browser e tecnologie lato client

La tecnologia lato client si avvale di un moderno browser con supporto ad HTML, CSS, e Javascript.

Con browser si intende quel particolare software che consente di visualizzare contenuti di pagine web e di interagire con essi permettendo così all'utente di navigare in internet browser vengono principalmente utilizzati su personal computer, ma anche su altri dispositivi che consentono la navigazione in internet, come i palmari e gli smartphone. Quelli più noti e diffusi sono Internet Explorer, Mozilla Firefox, Opera, Google Chrome e Safari. È importante conoscere l'architettura di un browser per capire meglio il funzionamento del framework e delle tecnologie utilizzate per lo sviluppo dell'applicazione.

Le architetture dei moderni browser risultano simili tra di loro e costante di 8 parti principali:

1. User Interface Subsystem: è il layer tra l'utente e il motore del browser, si può trovare integrato nel desktop environment oppure essere fornito come applicativo esterno
2. Il Browser Engine: è un componente "embeddable" che fornisce l'interfaccia ad



alto livello al motore di rendering, carica gli URL assegnati e supporta azioni di browsing primitive quali, "forward", "back", "refresh", ecc

3. **Rendering Engine:** il sottosistema che produce una rappresentazione visuale per l'URL. È capace di interpretare HTML e XML, opzionalmente correlati a CSS e a immagini. Calcola l'esatto layout della pagina e include il parser HTML. È il componente principale dell'architettura, e la sua ottimizzazione è la chiave per la velocità dell'intero software. I motori di rendering utilizzati nei principali browser sono riportati in tabella insieme alle principali caratteristiche.

Rendering Engine	Sviluppatore	Browser Principali	Sistema Operativo	Licenza
Trident	Microsoft Corporation	Microsoft Internet Explorer	MS Windows	Proprietaria
Gecko	Mozilla Foundation	Mozilla Firefox	MS Windows, Mac OS X, Unix, Linux, BSD	MPL/GNU GPL/GNU LGPL tri-license
Presto	Opera Software	Opera	MS Windows, Mac OS X, Unix, Linux, BSD, Symbian OS	Proprietaria
WebKit	Apple, Google, KDE, Nokia, Palm e altri	Google Chrome, Apple Safari, Apple Safari Mobile	MS Windows, Mac OS X, iPhone OS, Unix, Linux, BSD, Symbian OS	GNU LGPL, BSD-style

4. **Networking sub system:** implementa i protocolli di trasferimento file quali HTTP ed FTP. Traduce tra diversi set di caratteri e risolve i tipi MIME per identificare il tipo di file. Può implementare, inoltre una cache, per evitare il reperimento di risorse online recentemente reperate.
5. **Javascript Engine:** Il motore interprete di javascript; alcune funzionalità di Javascript possono essere disabilitate dal Browser Engine o dal Rendering Engine per motivi di sicurezza.
6. **Il parser XML:** analizza il documento e crea l'albero DOM accessibile tramite Javascript; questo componente è spesso derivato da parser già esistenti e ottimizzati. Fornisce vari livelli di supporto a codice HTML non standard.

7. Il BackEnd grafico: questo sottosistema fornisce le primitive per la visualizzazione del motore grafico in una finestra; solitamente le funzionalità sono a stretto contatto con il sistema operativo;
8. Il Data Persistence sub system: immagazzina i vari dati associati alla sessione quali cookie, segnalibri, certificati di sicurezza e memoria cache

3.4.1. HTML5

HTML è il linguaggio di base per la creazione di pagine web, con il quale viene definita la struttura e il contenuto della pagina. Tutti i siti web sono scritti in HTML, codice che viene letto per essere visualizzato dal browser, il quale genera la pagina che viene visualizzata sullo schermo del computer.

L'HTML non è un linguaggio di programmazione, ma un linguaggio di markup, ossia un linguaggio che descrive come deve essere trattato (in questo caso *visualizzato*) il contenuto, testuale e non, di una pagina web.

È un linguaggio di pubblico dominio la cui sintassi è stabilita dal World Wide Web Consortium (W3C), e che è basato su un altro linguaggio avente scopi più generici, l'SGML.

È stato sviluppato alla fine degli anni ottanta da Tim Berners-Lee al CERN di Ginevra. Verso il 1994 ha avuto una forte diffusione, in seguito ai primi utilizzi commerciali del web.

Nel corso degli anni, seguendo lo sviluppo di Internet, l'HTML ha subito molte revisioni, ampliamenti e miglioramenti, che sono stati indicati secondo la classica numerazione usata per descrivere le versioni dei software. Attualmente l'ultima versione disponibile è la versione 4.01, resa pubblica il 24 dicembre 1999. Dopo un periodo di sospensione, in cui il W3C si è focalizzato soprattutto sulle definizioni di XHTML (applicazione a HTML di regole e sintassi in stile XML) e dei fogli di stile (Cascading Style Sheet, CSS), nel 2007 è ricominciata l'attività di specifica con la definizione, ancora in corso, di HTML 5, attualmente allo stato di bozza, ma nonostante ciò, in parte supportata dai moderni browser.

Le novità introdotte da HTML5, sono molteplici:

- nuovi elementi di layout (tra cui i tag `<section>`, `<article>`, `<figure>`) per rendere più semplice la creazione di contenuti editoriali e della loro identificazione da parte dei motori di ricerca.
- i tag `<video>` e `<audio>` che permettono di incorporare filmati e elementi audio all'interno della pagina senza l'ausilio di plugin esterni.
- il tag `<progress>` che permette di visualizzare il suo contenuto quanto viene riconosciuto un certo evento nel sistema.
- il tag `<time>` per la rappresentazione di date.
- il tag `<canvas>` che permette di inglobare elementi di grafica bitmap, e la sua creazione dinamica, all'interno del documento. HTML Canvas verrà descritto in dettaglio nel paragrafo browser e grafica.

In questo lavoro, come meglio spiegato nel capitolo “*Grafica nei browser*” verrà impiegata il tag <canvas>. Si suppone di poter ampliare, inoltre, l’applicazione in futuro con l’inserimento del video nativo messe a disposizione dai nuovi tag di HTML 5.

3.4.2. Javascript

La programmazione lato client nel browser avviene tramite il linguaggio Javascript.

La caratteristica principale di Javascript è quella di essere un linguaggio interpretato. Il codice quindi non viene compilato bensì inviato ad un interprete (in Javascript lato client esso è incluso nel browser che si sta utilizzando) che esegue riga per riga, a *run-time*, quanto trascritto nello script. Javascript presenta quindi tutte le caratteristiche di un normale linguaggio interpretato (e di conseguenza i suoi vantaggi e svantaggi) con una sintassi analoga a quella di un linguaggio compilato (essa è relativamente simile a quella del C, del C++ e del Java)[7], quindi con la possibilità di utilizzare funzionalità tipiche dei linguaggi di programmazione ad alto livello (strutture di controllo, cicli, etc.) e con in più anche la potenzialità di definire strutture più complesse, vicine a quelle adottate nei normali linguaggi Object Oriented (creazione di prototipi, istanziazione di oggetti, costruttori).

Un'altra caratteristica importante di Javascript consiste nel suo essere un linguaggio detto "debolmente tipizzato", cioè il tipo delle variabili può non essere assegnato in fase di dichiarazione e le variabili stesse vengono interpretate e convertite in maniera automatica dall'interprete durante l'esecuzione.

Come nel C, il linguaggio non ha propri costrutti di input o output; mentre il C si affida alle librerie I/O standard, un interprete Javascript si basa su un programma ospite in cui è integrato. Ci sono molti programmi ospiti di questo tipo, di cui quelli relativi al Web (i Browser) sono gli esempi più noti.

JavaScript, se integrato in un browser Web, si collega tramite interfacce chiamate DOM (Document Object Model) alla struttura della pagina HTML, creando un potente sistema per generare e modificare programmaticamente gli elementi grafici della pagina web, quali immagini, testo, campi dati pulsanti ecc. Può, inoltre, essere utilizzato per compiere determinate operazioni sulla pagina non consentite da HTML statico, come ad esempio il controllo o la lettura di valori nei campi di input.

Un'altra funzione risultata utile per lo sviluppo è la valutazione delle espressioni regolari (introdotte nella versione 1.2, in Netscape Navigator 4 e Internet Explorer 4 e oggi largamente diffuse nei principali browser). Le espressioni Javascript contenute in una stringa possono essere valutate usando la funzione *eval*.

Sfortunatamente, gli standard DOM imposti dal W3C non sempre vengono rispettati dai vari browser: browser diversi (anche a seconda del loro motore di rendering) espongono diversi oggetti o metodi allo script (Internet Explorer è solito aderire agli standard con piccole modifiche, e tratta ad esempio l'oggetto event come globale; Opera non supporta le funzioni alert() e confirm()), ed è quindi spesso necessario implementare controlli aggiuntivi ad una funzione JavaScript, per garantirne la compatibilità con ciascun browser.

Al di fuori del Web, interpreti JavaScript sono integrati in diverse applicazioni. Adobe Acrobat e Adobe Reader supportano JavaScript nei file PDF. La piattaforma Mozilla, che è alla base di molti diffusi browser Web, usa JavaScript per implementare l'interfaccia utente e la logica di transazione dei suoi vari prodotti. Gli interpreti JavaScript sono integrati anche nelle applicazioni proprietarie prive di interfacce programmabili via script. Infine la tecnologia Windows Script Host di Microsoft supporta JavaScript (via JScript) un linguaggio di scripting per i sistemi operativi.

Ciascuna di queste applicazioni fornisce il proprio modello a oggetti che dà accesso all'ambiente ospite, con il nucleo del linguaggio JavaScript che rimane per lo più invariato in ciascuna applicazione.

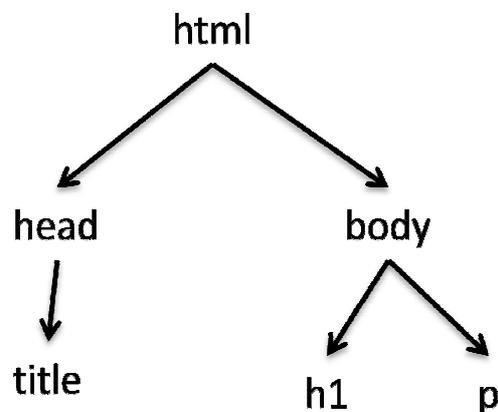
3.4.3. Document Object Model

Document Object Model è un'interfaccia linguaggio-neutrale e piattaforma-neutrale che permette a programmi e script di accedere e aggiornare dinamicamente la struttura e lo stile di un documento. DOM è lo standard utilizzato dai browser per la rappresentazione dei documenti strutturati relativi al layout della pagina HTML. DOM è inoltre la base per una vasta gamma delle interfacce di programmazione delle applicazioni. alcune di esse sono standardizzate dall'organismo W3C.

Come esempio il documento HTML:

```
<!-- Documento-->  
  
<html>  
  
<head>  
  
<title>Titolo</title>  
  
</head>  
  
<body>  
  
<h1>Titolo pagina</h1>  
  
<p>  
Paragrafo  
</p>  
</body>  
</html>
```

Viene interpretato dal parser XML del browser, il quale costruisce l'albero delle dipendenze che costituisce una rappresentazione logica della pagina web:



Le varie componenti grafiche codificate nell'albero risultano accessibili a Javascript tramite le interfacce DOM che il Javascript Engine mette a disposizione, permettendo la modifica dinamica dell'interfaccia.

DOM non pone limitazioni alla struttura dei dati di un documento, bensì permette di visualizzare un documento "ben formato" sotto forma di albero: la maggior parte dei parser XML (come Xerces) e i processori XSL (per esempio Xalan) sono stati sviluppati per usare la struttura ad albero. Questa implementazione richiede che l'intero contenuto di un documento venga analizzato e salvato in memoria. DOM è utilizzato principalmente per recuperare informazioni da documenti con una struttura non standard, cioè dove gli elementi devono essere trovati in modo casuale. Per le applicazioni basate su XML che usano un processo di lettura e scrittura per analisi, DOM presenta un grande spreco di memoria; per questo tipo di applicazioni esistono altri tipi di modello, come SAX, che permette l'accesso alla struttura del documento senza la necessità di caricarne l'intero contenuto in memoria.

3.4.4. Dojo

Per semplificare lo sviluppo e il supporto ai diversi browser è stata utilizzato il *framework Dojo* per lo sviluppo lato server dell'applicazione. A differenza di altri framework (come ad esempio JQuery) Dojo viene fornito con una serie di componenti grafici integrabili nel codice HTML tramite l'utilizzo di particolari attributi che si aggiungono agli usuali forniti dalla sintassi HTML. Inoltre fornisce funzionalità per la gestione di dati e per la comunicazione client-server tramite lo standard JSON.

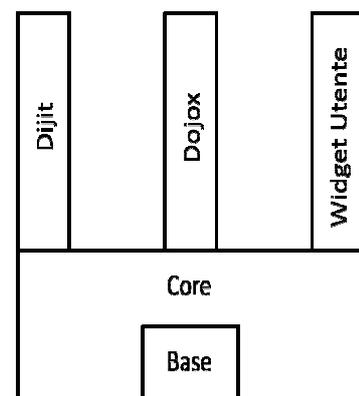
Il framework gratuitamente scaricabile dal sito della fondazione che lo mantiene (<http://www.dojotoolkit.org/>) è distribuito tramite doppia licenza BSD e AFL. È possibile inoltre avere accesso ad una ricca documentazione delle funzionalità fornite da Dojo.

3.4.4.1. *Architettura*

Il framework si suddivide in tre parti principali: Base e Core, Dijit e Dojox.

Base costituisce la parte kernel del framework e fornisce una libreria ultra compatta e ottimizzata di funzioni di base sia per l'utilizzo delle altre parti del framework, sia per la velocizzazione di compiti solitamente usati nello sviluppo web, quali la serializzazione degli oggetti Javascript in JSON, e la comunicazione web. Dojo Base è contenuta in un unico file "dojo.js" e, se per lo sviluppo si necessita solo delle funzioni di base supportate quella libreria è possibile includere nell'applicazione solo questo file.

Core, anch'essa contenuta nel file dojo.js contiene invece le funzionalità di base per il supporto dei namespace per il caricamento modulare dei vari widget e, e per le funzionalità di visualizzazione come internazionalizzazione, temi visivi, accessibilità ecc.



Dijit e *Dojox* costituiscono invece una ricca collezione di widget, ovvero elementi grafici in stile AJAX, come pulsanti, campi di input event-driven, layout, componenti per la gestione del calendario. La differenza sostanziale tra le due parti è il supporto che la fondazione Dojo fornisce al contenuto dei due toolkit.

In particolare Dijit contiene widget che sono considerati stabili dalla Dojo foundation e rispettano determinate caratteristiche di standardizzazione rispetto alle funzionalità fornite da Core. Per questi plug-in viene fornita una buona documentazione on-line. Dojox contiene, invece, i plug-in che, benché anch'essi considerati stabili e utilizzabili per progetti di sviluppo non presentano ancora il pieno supporto a tutte le funzionalità fornite da Core. Per questi non viene fornita una adeguata documentazione, però in rete è possibile reperire moltissime informazioni tramite il forum mantenuto dalla fondazione stessa.

Al toolkit fornito di base da Dijit e Dojox è possibile aggiungere ulteriori widget tramite il meccanismo fornito da Core per lo sviluppo dei widget inseriti nella libreria.

A differenza di Base e Core gli elementi contenuti in Dijit e Dojox sono divisi in molti file e occupano in tutto diversi megabyte di memoria. Per evitare la perdita di prestazioni dovuta al caricamento dell'intero toolkit Base fornisce un meccanismo per il caricamento dei soli widget necessari all'applicazione.

L'utilizzo del framework è molto semplice.

Per prima cosa è necessario inserire nel codice HTML i tag di caricamento della libreria:

```
<script type="text/javascript" src="/dojo/dojo.js"
djConfig="parseOnLoad:true"></script>
```

All'usuale tag `<script>` da utilizzare per il caricamento è possibile aggiungere l'attributo `djConfig="parseOnLoad:true"` utilizzato da dojo per la lettura dei parametri di configurazione

Si può dunque subito iniziare a programmare, creando uno script nell'header della pagina:

```
<script type="text/javascript">
/* contenuto Javascript, usando Dojo */
</script>
```

Come precedentemente descritto dojo base introduce alcune funzionalità e alcuni miglioramenti alle istruzioni fornite da javascript che ne migliorano la compatibilità cross-browser.

3.4.4.2. *Onload()* e *Dojo.addOnLoad()*

Dovendo utilizzare la libreria per interagire con i componenti della pagina ci si trova di fronte ad un problema: JavaScript viene infatti interpretato insieme all'HTML, e ciò significa che se il codice si trova nell'header, esso verrà eseguito prima ancora che gli oggetti della pagina siano stati creati, impedendone ovviamente la manipolazione.

L'unica soluzione sarebbe dunque di porre tutti gli script al termine del file o usare l'oggetto JavaScript onLoad, rendendo però il tutto decisamente meno chiaro e ordinato.

Per risolvere il problema Dojo Base offre un semplice metodo: `dojo.addOnLoad()`, al quale, come parametro va passata la funzione da eseguire ad avvenuto caricamento della pagina.

E' possibile modificare in questo modo il tag `<script>` incorporando al suo interno il comando:

```
<script type="text/javascript">
dojo.addOnLoad(function(){
    //Funzione da eseguire
});
</script>
```

3.4.4.3. Accesso ai nodi DOM

Per avere accesso all'albero della gerarchia DOM javascript e la maggior parte dei browser moderni utilizza il comando "`document.getElementById()`", al quale va passato come parametro una stringa contenente l'ID del nodo. Tramite il comando si ottiene un puntatore all'oggetto tramite il quale è possibile interagire con il nodo vero e proprio e la sua visualizzazione grafica (se esiste).

Tramite la libreria dojo Base è possibile accedere sia all'albero DOM, con un supporto migliorato cross-browser, sia ai widget della libreria dijit e dojox presenti nella pagina tramite il comando `dojo.byId()`, per l'albero DOM e `dijit.byId()` per l'accesso ai widget.

3.4.4.4. Eventi

Un netto miglioramento alla gestione degli eventi si ottiene tramite l'utilizzo dell'API messa a disposizione da dojo. In HTML gli eventi vengono distinti in due categorie. La prima categoria comprende gli eventi globali, eventi cioè associati alla sessione del browser, come `OnLoad` lanciato al completo caricamento della pagina. Una seconda categoria contiene invece gli eventi collegati ai nodi DOM come ad esempio il click su un pulsante. La gestione di questi eventi risulta spesso problematica in quanto non tutti i browser considerano nella stessa categoria uno stesso evento: ad esempio l'evento `keypressed` è considerato da alcuni browser legato alla sessione, da altri legato al nodo DOM e da altri ancora utilizzabile in entrambi i modi. Altri eventi come ad esempio `doubleclick` non sono supportati da tutti i browser

Per gestire gli eventi, Dojo usa un metodo estremamente versatile: `dojo.connect()`, che accetta come argomenti l'ID del nodo, il tipo di evento e la funzione da eseguire, per gli eventi legati ai nodi; `dojo.subscribe()` che accetta come parametri la stringa contenente il tipo di evento e la funzione da eseguire, per gli eventi globali. La compatibilità cross browser dei vari eventi, benché più limitata (dojo Connect non accetta alcuni tipi di evento come ad esempio `DOMActivate`, che comunque risultano poco usati), è sicuramente migliore.

Il codice da utilizzare per l'inserimento degli eventi è:

```
<script type="text/javascript">
dojo.addOnLoad(function(){
//Il nodo da modificare:
nodo = dojo.byId("nodoTest");
//il metodo da eseguire:
function cambiaContenuto(evento){
evento.currentTarget.innerHTML = "Nuovo contenuto";
}
//connettiamo nodo, evento e funzione con dojo:
dojo.connect(nodo, "onClick", cambiaContenuto);
dojo.subscribe("globalEvents", globalGuy);
});
</script>
```

Associare un metodo ad un evento si rivela estremamente utile, specialmente nello sviluppo di applicazioni Web, poiché essi ricalcano un meccanismo simile a quello dei listeners di Java.

Se poi si considera che Dojo passa al metodo invocato l'oggetto event, il meccanismo si rivela utile. Tale oggetto infatti contiene al suo interno tutte le informazioni relative all'evento: in quale ramo dell'albero esso è avvenuto, le coordinate del mouse, il tasto premuto (in caso di evento "onKeyPress", relativo agli eventi di tastiera) e altri metodi utili. Se per esempio si volessero visualizzare le coordinate del mouse, basterà inserire nel metodo invocato questa riga:

```
alert("Il mouse è alla posizione " + evento.pageX + "," + evento.pageY);
```

3.4.4.5. JSON

Come verrà spiegato in dettaglio nel capitolo "JSON" per la trasmissione dei dati verrà utilizzata la codifica JSON, uno standard per la trasmissione di stringhe ormai molto usato nelle applicazioni web.

Tramite Dojo è possibile utilizzare un semplice comando per la serializzazione e deserializzazione di stringhe formattate in JSON.

Il comando `dojo.fromJson(/*stringa*/)` riceve in input una stringa formattata in nello standard JSON e ritorna l'oggetto rappresentato dalla stringa in un formato perfettamente utilizzabile da Javascript.

Il comando `dojo.toJson(/*oggetto*/)` riceve come argomento un qualsiasi oggetto Javascript (ad eccezione dei nodi DOM) e lo serializza in standard DOM.

Ad esempio il codice:

```
var obj = { a:"one", b:3, c:true };  
var data = dojo.toJson(obj);  
  
ritorna la stringa:  
  
'{"a":"one","b":3,"c":true}'
```

Questo tipo di funzionalità non è inclusa nell'API standard di Javascript.

3.4.4.6. Comunicazione client-server

Tramite dojo è possibile utilizzare un semplice comando per la comunicazione client/server che va a sostituire la complessa, sebbene più completa, API fornita da javascript per la comunicazione asincrona con il web server.

Il comando fornito da Dojo è riportato nel seguente esempio:

```
dojo.xhrPost({  
  url: "./serverside/matrix.py/get_stats",  
  handleAs: "text",  
  content: {  
    initial: mainSession.matrix.getMatrix()  
  },  
  load: function(response, args){  
    update_globalchart(dojo.fromJson(response))  
  },  
  error: function(response, args){  
    //alert(response);  
    return response;  
  }  
});
```

il comando `dojo.post` richiede come parametro in ingresso un unico oggetto javascript con le seguenti proprietà:

- `url`: l'url del servizio web da contattare
- `handleAs`: il tipo di informazione ricevuto, in questo caso testuale
- `content`: i parametri da passare al servizio; in questo caso la funzione `mainSession.matrix.getMatrix()` ritorna una stringa di testo formattata tramite JSON
- `load`: la funzione da eseguire al momento della ricezione dei dati
- `error`: la funzione da eseguire qualora ci siano stati problemi nella comunicazione

3.4.5. CSS: Cascading Style Sheets

I *fogli di stile*, meglio noti con l'acronimo CSS (dall'inglese Cascading Style Sheet), vengono utilizzati per definire la rappresentazione grafica di documenti HTML, XHTML e XML. Da sempre infatti, nelle intenzioni dell'organismo di standardizzazione del web W3C, HTML, così come la sua recente evoluzione, XHTML, dovrebbe essere visto semplicemente come un linguaggio strutturale, esente da qualunque scopo attinente la presentazione di un documento. Per questo obiettivo, ovvero arricchire l'aspetto visuale ed estetico di una pagina, lo strumento designato sono appunto i CSS. L'idea alla base è quella della separazione del contenuto della pagina dalla sua presentazione. Questa separazione permette di adattare, utilizzando una struttura HTML definita, la pagina web (o applicazione) a diverse risoluzioni dello schermo tipiche dei vari devices. Le caratteristiche dei CSS permettono, inoltre, un aggiornamento veloce alla componente grafica della pagina. Le regole per comporre i fogli di stile sono contenute in un insieme di direttive (Recommendations) emanate a partire dal 1996 dal W3C.

Il fogli CSS collegati ad una pagina web possono essere utilizzati per modificare moltissime caratteristiche della grafica della pagina, come la spaziatura dei caratteri, i bordi, gli sfondi, la posizione dei componenti, migliorando notevolmente le caratteristiche presenti in HTML.

La prima specifica ufficiale di CSS (CSS1) risale al dicembre del 1996. Nel maggio 1998 è stata la volta della seconda versione: CSS2. Niente stravolgimenti, ma molte aggiunte rispetto alla primatra cui valori di proprietà e definizioni per stili non canonici come quelli rivolti alla stampa o alla definizione di contenuti audio. È attualmente allo stato di bozza la nuova specifica CSS3 di cui però alcuni Rendering Engine godono già del pieno supporto (WebKit e Presto). Tra le varie migliorie introdotte CSS3 introduce il supporto ai layer 3D.

3.4.6. Grafica nei browser

È possibile inserire elementi grafici, quali animazione e immagini vettoriali, all'interno di pagine web attraverso varie soluzioni. Alcune di queste non sono integrate nel browser ma sono utilizzabili comunque all'interno di essi in moltissime piattaforme tramite l'utilizzo di plug-in quali Adobe Flash e Microsoft Silverlight. Questo tipo di soluzione comporta però lo svantaggio di utilizzare delle tecnologie proprietario e non accessibili per qualsiasi dispositivo (ad esempio Apple iOS non supporta né Flash, né Silverlight).

Recentemente sono stati, però sviluppati degli standard pubblici, supportati nativamente dai browser e integrati nei vari Rendering Engine. Per motivi commerciali, queste tecnologie sono soggette sempre più ad ottimizzazioni per alcuni motori sono in fase di sviluppo delle versioni che permetteranno di renderizzare la tramite GPU (cosa che ancora non avviene nelle soluzioni proprietarie). È inoltre possibile un maggior controllo delle componenti grafiche utilizzando DOM e Javascript: infatti tutti gli elementi grafici integrati nel codice HTML sono accessibili tramite l'albero DOM della pagina da Javascript.

Le tecnologie incorporate nei browser sono HTML5 Canvas, SVG e VML. In tabella sono mostrate le tecnologie supportate dai vari motori di Rendering

Rendering Engine	HTML5 Canvas	SVG	VML
Trident (IE)	(tramite exCanvas)		X
Gecko (Mozilla)	X	X	
Presto (Opera)	X	X	
WebKit (Safari, Chrome)	X	X	

3.4.6.1. HTML5 Canvas

L'elemento canvas, introdotto dalle nuove specifiche di HTML 5, fornisce supporto per in canvas, cioè una porzione di schermo nella quale può essere renderizzata grafica "on-the-fly" per tabelle, grafici, giochi ecc. È importante sottolineare come questo elemento non sia stato pensato per disegnare all'interno della pagina elementi grafici statici come intestazioni o immagini bitmap, oppure elementi della pagina per i quali esistono altri tipo di soluzioni (come semplici bottoni oppure campi di input). È invece molto utile e molto performante quando viene utilizzato per animazioni o per effetti grafici dinamici complessi. In futuro si stanno inoltre sviluppando soluzioni per l'utilizzo della grafica 3D.

Il tag <canvas> che può essere inserito all'interno di qualsiasi elemento <div> della pagina ha due attributi (width e height) che definiscono le dimensioni della porzione di schermo da utilizzare per il rendering dinamico accessibile anche tramite foglio di stile. È possibile inoltre definire una dimensione standard per il canvas e modificarla poi tramite CSS per adattarlo alla dimensione della pagina: questo permette di elaborare la grafica ad una risoluzione ben definita che verrà poi scalata senza bisogno di ulteriore codice (oltre alla modifica del foglio CSS).

L'inserimento dell'elemento può avvenire tramite HTML con il comando:

```
<canvas id="element_ID" width="150" height="150"></canvas>
```

Per reperire l'elemento canvas dall'albero DOM si può utilizzare il comando:

```
var canvas = document.getElementById('tutorial');
```

```
var ctx = canvas.getContext('2d');
```

È possibile inoltre verificare il supporto del browser al canvas tramite Javascript:

```
var canvas = document.getElementById('tutorial');
```

```
if (canvas.getContext){
```

```
    var ctx = canvas.getContext('2d');
```

```
    // drawing code here
```

```
} else {
```

```
    // canvas-unsupported code here }
```

Una volta inserito il `<canvas>` è possibile inserire vari tipi di elemento (immagine, path, ellissi, archi e curve di Bezier), ai quali è possibile applicare rotazioni, traslazioni, ridimensionamenti, bordi ecc. è inoltre possibile applicare animazioni e gradienti di colore alle varie forme inserite comporle a creare forme complesse ed effettuare operazioni booleane su di esse.

È ormai supportato da quasi tutti i browser. L'unica eccezione degna di nota è Internet Explorer il quale necessita di una libreria Javascript aggiuntiva (ExCanvas reperibile al sito <http://excanvas.sourceforge.net/>) che va caricata all'avvio della pagina. Le performance sono comunque molto scarse. Per i prossimi mesi è prevista una release (la 9) che dovrebbe introdurre il supporto a HTML5 Canvas nativo su questo browser.

3.4.6.2. SVG

Scalable Vector Graphics abbreviato in SVG, indica una tecnologia in grado di visualizzare oggetti di grafica vettoriale e, pertanto, di gestire immagini di dimensioni scalabili. Più specificamente si tratta di un linguaggio derivato dall'XML, che si pone l'obiettivo di descrivere figure bidimensionali statiche e animate. La sintassi è quella tipica del linguaggio di markup, e può essere inglobato all'interno del codice HTML tramite il comando:

```
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg"></svg>
```

Al cui interno verranno inseriti i vari elementi grafici (sempre tramite tag) che posso essere:

Forme (`<Rectangle>`, `<Circle>`, `<Ellipse>`, `<Line>`, `<Polygon>`, `<Polyline>`, `<Path>`) Filtri e Gradienti (`<Gaussian>`, `<Linear>`, `<Radial>`). Gli elementi risultano accessibili dall'albero DOM e quindi utilizzabili da Javascript. È possibile inoltre utilizzare SVG come linguaggio per l'interscambio di immagini vettoriali: occorre in questo caso salvare in un file formattato secondo lo standard SVG la struttura del documento.

Tali sono integrabili in una pagina HTML all'interno, ad esempio di un `<iframe>`:

```
<iframe src="nomefile.svg" width="300" height="100"></iframe>
```

3.4.6.3. VML

Come SVG, VML (Vector Markup Language) è un linguaggio derivato da XML, destinato alla creazione di grafica vettoriale in 2D e 3D su pagine web e non. VML è stato sottomesso come proposta standard al W3C nel 1998 da Autodesk, HP, Macromedia e Microsoft in concorrenza con PGML proposto da Adobe e Sun Microsystems. Dopo un lungo esame, il W3C decide di fondere i due formati dando luce al formato SVG.

Il linguaggio VML può integrarsi sia direttamente nel codice HTML, sia tramite l'elemento `<v:vlframe>` legato ad un o più file XML. Questo elemento permette di sistemare le grafica vettoriali in modo analogo alle immagini bitmap. L'utilizzazione degli attributi `width` e `height` permettendo di ridimensionare gli elementi grafici senza perdita di qualità. Numerosi effetti di filtro

sono possibili sulle immagini e l'extrusion 3D permette la messa in rilievo di ogni elemento VML con la possibile aggiunta di effetti di luce apportando profondità e realismo alle grafica.

Anch'esso è un linguaggio che permette realizzazioni complesse su file di estensione ".vml".È, inoltre, maneggiabile con JavaScript.

Una applicazione che usa VML è Google Maps per il reso vettoriale con Internet Explorer. La stessa applicazione utilizza, invece, SVG per i browser che lo supportano.

3.4.6.4. La libreria Raphaël

Per aggirare le limitazioni al supporto delle tre tecnologie da parte dei browser è stata utilizzata una libreria di funzioni Javascript che mette a disposizione un'API di programmazione che si interfaccia con le diverse tecnologie a seconda del motore di rendering nella quale si trova ad operare.

Raphaël è una piccola libreria Javascript nata per semplificare il lavoro con la grafica vettoriale nella programmazione web che permette ad esempio di creare grafici oppure elaborare le immagini (con traslazioni, ridimensionamenti ecc.). Essa utilizza le specifiche SVG e VML come base per la creazione della grafica e la renderizza, dove possibile, all'interno di un canvas.

La libreria attualmente supporta Firefox, Safari, Opera e Internet Explorer.

Il suo funzionamento è molto semplice. È necessario innanzitutto creare, tramite il tag HTML <div>, la sezione di pagina nella quale si vuole inserire l'area grafica:

```
<div id="networkSurface"></div>
```

È possibile a questo punto creare l'oggetto *Raphael*, tramite Javascript, che rappresenta l'oggetto canvas o l'area nella quale si andrà a disegnare. La sua definizione può avvenire in due modi:

- `var paper = Raphael(10, 50, 320, 200);`
questo comando assegna alla variabile *paper*, un oggetto Raphael e riserverà la porzione di schermo compresa tra le coordinate inserite all'area grafica (in questo caso il comando <div> non serve)
- `var paper = Raphael(document.getElementById("networkSurface"), 320, 200);`
oppure
`var paper = Raphael("networkSurface", 320, 200);`
questi due comandi creano l'area grafica all'interno della sezione <div> con id=networkSurface; la differenza tra i due comandi sta nel modo a cui si accede all'albero DOM; nel primo caso, viene passato il nodo DOM direttamente, nel secondo il comando riceve in input il parametro che corrisponde alla stringa contenente l'id del nodo. Il comando si occupa di reperire il nodo DOM e di assegnargli l'oggetto grafico.

Poiché, per essere creato, l'area grafica necessita dell'area <div> (nel secondo caso) è necessario accertarsi che, al momento dell'esecuzione del codice Javascript, la porzione di albero DOM necessaria sia stata correttamente caricata in memoria. Per verificare ciò è possibile utilizzare un comando messo a disposizione dalla libreria standard di Javascript presente in ogni browser:

```
Var paper = false  
  
function init() {  
  
paper = Raphael(document.getElementById("networkSurface"), 320, 200);  
  
}  
  
window.onload = init;
```

tramite il comando `window.onload` è possibile eseguire una determinata funzione nel momento in cui l'intero albero DOM è stato caricato in memoria, cosicché il codice possa avere accesso all'intera struttura della pagina web. Per fare ciò è necessario definire prima la funzione da utilizzare (in questo caso `init()`) e successivamente sostituire la funzione associata a `window.onload` con la nuova funzione `init()` (tramite `window.onload = init`).

Per il disegno vero e proprio la libreria fornisce un set di comandi per la creazione di forme, testi rasterizzato e per la gestione di immagini, accessibili tramite l'oggetto Raphael associato all'area grafica creata. Alcuni dei comandi utilizzabili sono:

- `var img = paper.image("apple.png", 10, 10, 80, 80);`
inserisce un'immagine, con `path="apple.png"`, alle coordinate indicate
- `var c = paper.circle(50, 50, 40);`
inserisce una circonferenza con centro e raggio indicati
- `var t = paper.text(50, 50, "testo da inserire");`
inserisce il testo indicato con centro nel punto `P(50, 50)`
- `var pt = paper.path("M10 10L90 90");`
inserisce una polilinea; per la sua definizione è possibile utilizzare la codifica definita per i documenti SVG;

è possibile poi definire insiemi di forme per poter operare su di loro con una unica trasformazione:
`var st = paper.set();`

una volta definito il set, è possibile inserire forme al suo interno con il comando:

```
st.push(  
  
    paper.circle(10, 10, 5),  
  
    paper.circle(30, 10, 5)  
  
);
```

le trasformazioni applicabili ai vari oggetti inseriti sono traslazioni, rotazioni, e ridimensionamenti. Esse sono accessibili tramite la variabile che rappresenta la forma inserita, ad esempio:

```
var c = paper.circle(10, 10, 10);  
  
c.rotate(45);
```

permette di eseguire su un oggetto `circle` creato, e accessibile tramite la variabile `c`, una rotazione attorno al proprio asse di 45° .

Infine è possibile modificare gli attributi come colore, sfondo, spessore della linea ecc. ed effettuare delle animazioni sulle forme utilizzate, ad esempio:

```
var c = paper.circle(10, 10, 10);  
c.attr({fill: "#000", stroke: "#f00", opacity: 0.5});  
c.animate({fill:#fcc}, 2000);
```

in questo esempio viene prima creato un oggetto di tipo circonferenza sul quale lavorare, successivamente vengono modificati alcuni suoi attributi con la funzione `c.attr()` e poi viene animato con la funzione `c.animate()`. Il funzionamento di `attr` e `animate` è molto semplice: come parametro va passato un oggetto Javascript con le caratteristiche da definire (nel primo caso) oppure con il valore finale che gli attributi devono avere alla fine dell'animazione (nel secondo caso). Alla funzione `animate` va inoltre passato un secondo parametro che indica il tempo in millisecondi della durata dell'animazione.

La libreria Raphael permette di gestire gli eventi legati alle varie forme definite, permettendo di creare contenuto interattivo all'interno dell'elemento grafico. Per associare una forma a un evento è necessario utilizzare uno dei seguenti comandi (a seconda del tipo di evento da associare):

```
element.click(function (event) { /*codice di gestione*/ });  
element.dblclick(function (event) { /*codice di gestione*/ });  
element.mousedown(function (event) { /*codice di gestione*/ });  
element.mousemove(function (event) { /*codice di gestione*/ });  
element.mouseout(function (event) { /*codice di gestione*/ });  
element.mouseover(function (event) { /*codice di gestione*/ });  
element.mouseup(function (event) { /*codice di gestione*/ });  
element.hover(function (event) { /*codice di gestione*/ }, function (event) { /*codice di gestione*/ });
```

All'ultimo evento devono essere passate due funzioni: la prima verrà eseguita all'entrata del mouse nella regione di schermo che contiene la forma associata, la seconda all'uscita. Per eliminare l'associazione forma/evento è necessario utilizzare lo stesso comando preceduto dal suffisso *"un"*, ad esempio:

```
element.unclick(f);
```

3.5. Programmazione lato server

Per sviluppo del software lato server sono state utilizzate varie tecnologie open source il cui utilizzo è consolidato nella pratica dello sviluppo web lato server.

In particolare è stato scelto utilizzando la metologia LAMP Apache come Web Server e Mysql come motore per il database. È stato invece utilizzato python come linguaggio di scripting lato server in luogo di PHP, perché il supporto al calcolo matriciale, necessario per l'elaborazione delle informazioni in tempo reale sulle sessioni di lavoro vanta di librerie funzionali e performanti come NumPy. Lo sviluppo è avvenuto su piattaforma Windows anche se il software è stato pensato per l'utilizzo su server con sistema Linux, per la migliore compatibilità delle tecnologie utilizzate (Apache, Mysql, Python) con l'ambiente operativo.

3.5.1. Motore di Database / Mysql

MySQL è un Relational DataBase Management System (RDBMS), composto da un client con interfaccia a caratteri e un server. Supporta la maggior parte della sintassi SQL e si prevede in futuro il pieno rispetto dello standard ANSI.

MySQL, essendo scritto in linguaggio C e C++, è disponibile su molti differenti sistemi operativi tra cui AIX, AmigaOS, BSDi, Digital Unix, FreeBSD, HP-UX, GNU/Linux, Mac OS X, NetBSD, Novell NetWare, OpenBSD, OS/2 Warp, SGI IRIX, Solaris, SunOS, SCO OpenServer, SCO UnixWare, SGI Irix, Tru64, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows 2003, Windows Vista. Le piattaforme di riferimento sono comunque Linux e Solaris. Sono disponibili dei driver per i linguaggi C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, Tcl e per le piattaforme Mono e .Net.

Infine, il linguaggio SQL di MySQL comprende numerose estensioni che sono tipiche di altri DBMS, quali PostgreSQL, Oracle e Sybase. In questo modo le query non standard scritte per altri DBMS in alcuni casi funzioneranno senza problemi.

Il codice di MySQL venne sviluppato fin dal 1979 dalla ditta TcX ataconsult, poi rinominata MySQL AB, ma è solo dal 1996 che viene distribuita una versione che supporta SQL, prendendo spunto da un altro prodotto: mSQL. MySQL AB è stata rilevata da Sun Microsystems nel 2008, mentre nel 2010 quest'ultima è stata acquisita da Oracle.

In MySQL una tabella può essere gestita da diversi storage engine, il componente di un DBMS che si occupa della creazione e del mantenimento delle tabelle nel disco. Ogni storage engine presenta proprietà e caratteristiche differenti (transazionale o meno, migliori prestazioni, diverse strategie di locking, funzioni particolari, ecc). Esiste poi un'API che si può utilizzare per creare in modo relativamente facile un nuovo storage engine, che poi si può installare senza dover ricompilare o riavviare il server.

Lo storage engine utilizzato in questo lavoro è InnoDB Poiché il suo lock per quanto riguarda i comandi SELECT è di tipo non-locking, caratteristica utile in sistemi multiutente. InnoDB offre delle

ottime performance in termini di prestazioni e utilizzo della CPU specialmente quando si ha a che fare con una grande quantità di dati. InnoDB può interagire tranquillamente con tutti gli altri tipi di tabelle in MySQL.

È stato, inoltre, scelto questo engine per le sue ottime prestazioni anche con grandi moli di dati e per la presenza dell'integrità referenziale (funzionalità non presente nell'engine standard, MyIsam).

3.5.2. HTTP server (Apache)

Apache HTTP Server, è il nome dato alla piattaforma server Web modulare più diffusa, in grado di operare da sistemi operativi Linux-like e Microsoft.

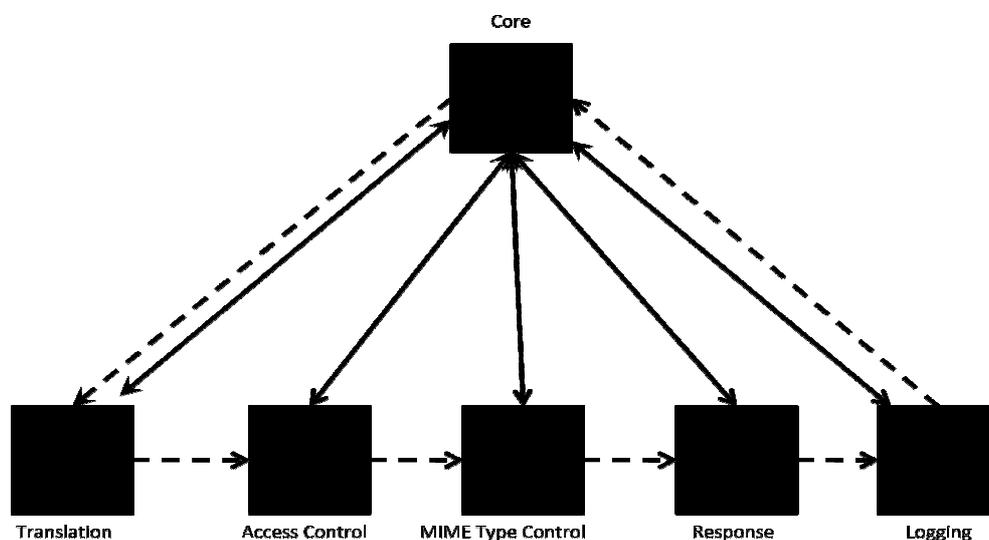
Il progetto Apache nacque nel 1995 sulla base del daemon HTTP pubblico sviluppato da Rob McCool al NCSA (National Center for Supercomputing Application).

La versione 1.0 fu pubblicata l'1 dicembre 1995. Nel giro di un anno, la sua diffusione aveva già superato quella del server NCSA da cui era derivato. La versione 2.0 di Apache venne rilasciata nel marzo 2000.

Operativamente, è composto da un demone, in ambiente UNIX, o da un servizio, in ambiente Microsoft, che sulla base delle impostazioni contenute nel file di configurazione httpd.conf permette l'accesso a uno o più siti, gestendo varie caratteristiche di sicurezza e potendo ospitare diverse estensioni per pagine *dinamiche*, come PHP, Jakarta/Tomcat o Python.

3.5.2.1. *Architettura apache*

Il Web Server apache presenta un'architettura modulare, quindi ad ogni richiesta del client, vengono svolte funzioni specifiche da ogni modulo di cui è composto, come unità indipendenti. Ciascun modulo si occupa di una funzionalità, ed il controllo è gestito dal core.



I moduli di Apache e la loro configurazione

I moduli:

1. Core: programma principale composto da un ciclo sequenziale di chiamate ai moduli.
2. Translation: traduce la richiesta del client.
3. Access Control: controlla eventuali richieste dannose.
4. MIME Type: verifica il tipo di contenuto.
5. Response: invia la risposta al client e attiva eventuali procedure.
6. Logging: tiene traccia di tutto ciò che è stato fatto.

Il core suddivide la richiesta ai vari moduli in modo sequenziale, usando i parametri di uscita di un modulo come parametri di accesso per l'altro, creando così l'illusione di una comunicazione orizzontale fra i moduli (Pipeline software).

Gli amministratori del server possono usare il file `httpd.conf`, che su sistemi unix solitamente è messo sotto `/etc/httpd/conf`, mentre su sistemi Windows è situato nella subdirectory `conf` della directory di installazione di Apache. Questo file offre tutta la libertà che offre il server, quindi aggiungere moduli, estensioni, nuovi mime-type e altro ancora. Per esempio se si volesse aggiungere un modulo bisognerebbe usare questa sintassi:

```
LoadModule nome_modulo percorso_del_file
```

I file `.htaccess`, presenti nelle singole cartelle contenenti il codice delle pagine web, permette una ulteriore personalizzazione del webserver a livello di directory. È utilizzato in particolar modo nell'hosting condiviso, per modificare le impostazioni standard fornite dal server stesso.

3.5.3. Python

Il linguaggio

Python è un linguaggio di programmazione ad alto livello, orientato agli oggetti, adatto, tra gli altri usi, per sviluppare applicazioni distribuite, scripting, computazione numerica e system testing. Ideato da Guido van Rossum all'inizio degli anni novanta, Python prende il nome dalla serie televisiva Monty Python's Flying Circus.

Python è un linguaggio multi-paradigma, che permette di scrivere programmi seguendo il paradigma object oriented, la programmazione strutturata oppure la programmazione funzionale. Il controllo dei tipi viene eseguito al runtime.

L'interprete python usa un garbage collector per la gestione automatica della memoria.

Python viene in genere considerato un linguaggio interpretato. In realtà però il codice sorgente non viene direttamente convertito in linguaggio macchina, ma passa prima da una fase di pre-compilazione in bytecode; lo stesso bytecode viene quasi sempre riutilizzato dopo la prima esecuzione del programma, evitando così di dover ogni volta interpretare il sorgente ed incrementando di conseguenza le prestazioni. Inoltre è possibile distribuire programmi Python

direttamente in *bytecode*, saltando la fase di interpretazione da parte dell'interprete nel PC dell'utilizzatore finale, e permettendo anche di avere programmi Python a sorgente chiuso.

Nonostante python sia classificato come linguaggio di scripting, la grande quantità di librerie disponibili e la facilità con cui questo linguaggio permette di scrivere software modulare favoriscono anche lo sviluppo di applicazioni molto complesse.

3.5.3.1. *Tipi di dati e strutture*

Python ha un gran numero di tipi base. Oltre ai tipi interi e floating point, supporta numeri interi arbitrariamente grandi e numeri complessi. Dalla versione 2.4 sono disponibili anche i numeri decimali (decimal), ovvero numeri con la virgola a precisione illimitata, che non soffrono di problemi di arrotondamento e stabilità tipici dei numeri floating point classici.

Supporta tutte le operazioni classiche sulle stringhe e, come avviene in Java e Javascript le stringhe sono oggetti immutabili, cosicché qualsiasi operazione che in qualche modo potrebbe alterare una stringa (come ad esempio la sostituzione di un carattere) restituirà invece una nuova stringa.

Essendo Python "debolmente tipizzato" (come Javascript), tutte le variabili sono in realtà semplici puntatori ad oggetto (reference), sono gli oggetti invece ad essere dotati di tipo. Ad esempio ad una variabile cui era assegnato un intero, un istante dopo può essere assegnata una stringa o un array. Sarà compito dell'interprete, a runtime, l'interpretazione e l'assegnazione del tipo alla variabile.

3.5.3.2. *Sintassi*

Python è stato progettato in modo da essere altamente leggibile. Visivamente si presenta in modo molto semplice e ha pochi costrutti sintattici rispetto a molti altri linguaggi strutturati.

Per esempio, Python ha solo due forme di ciclo: `for` che itera sugli elementi di una lista o su di un iteratore (equivalente al `foreach` di altri linguaggi) e `while` che itera fin tanto che l'espressione booleana indicata risulterà vera. Allo stesso modo ha solamente il costrutto `if...elif...else` per le scelte condizionate, ma non `switch` oppure `goto`.

Un aspetto inusuale del Python è il metodo che usa per delimitare i blocchi di programma, che lo rende unico fra i linguaggi più diffusi.

Nei linguaggi derivati dall'ALGOL – come Pascal, C e Perl – i blocchi di codice sono indicati con le parentesi oppure con parole chiave (il C ed il Perl usano `{ }`; il Pascal usa `begin` ed `end`). In questi linguaggi è solo una convenzione degli sviluppatori il fatto di indentare il codice interno ad un blocco, per metterlo in evidenza rispetto al codice circostante.

Python, invece, deriva il suo sistema di indentazione dal meno noto linguaggio di programmazione Occam: invece di usare parentesi o parole chiave, usa l'indentazione stessa per indicare i blocchi nidificati (si può usare sia una tabulazione, sia un numero arbitrario di spazi bianchi, ma lo standard Python è di 4 spazi bianchi). L'esempio che segue chiarisce questo aspetto; sono mostrate la versione C e Python di funzioni per il calcolo del fattoriale di un intero.

Un altro punto di forza di Python è la disponibilità di elementi che facilitano la programmazione funzionale. Come ci si può aspettare, questo rende ancora più comodo operare con liste o altri tipi contenitore.

3.5.3.3. *Libreria standard*

Python ha una vasta libreria standard, il che lo rende adatto a molti impieghi. Oltre ai moduli della libreria standard se ne possono aggiungere altri scritti in C oppure in Python stesso per soddisfare le proprie esigenze particolari. Tra i moduli già disponibili ve ne sono per scrivere applicazioni web tra cui il supporto a MIME, HTTP e a molti altri standard di Internet. Sono disponibili anche moduli per creare applicazioni con interfaccia grafica, per connettersi a database relazionali, per usare le espressioni regolari.

La libreria standard è uno dei punti forti di Python. Essa infatti è compatibile con tutte le piattaforme, ad eccezione di poche funzioni, segnalate chiaramente nella documentazione come specifiche di una piattaforma particolare.

3.5.3.4. *Python per il web: mod_python*

Apache integra, nella sua distribuzione standard, la possibilità di eseguire codice python dinamico tramite la tecnologia CGI. Tuttavia esistono diversi moduli di estensione per il web server Apache che ne migliorano considerevolmente le prestazioni. Il modulo scelto per questo lavoro è mod_python, scaricabile con licenza free dal sito <http://www.modpython.org/>.

Mod_python è un modulo per Apache che mette a disposizione vari *handler*, cioè, funzioni che vengono eseguite durante una delle fasi di elaborazione della richiesta utente da parte di Apache, grazie ai quali è possibile eseguire codice Python dall'interno del server in uno qualsiasi dei sette moduli descritti precedentemente. Ad esempio è possibile utilizzare del codice python alla richiesta del client, per restituire una pagina HTML formattata dinamicamente, oppure del codice XML o JSON ricavato dall'elaborazione dei dati presenti in un database. Può essere inoltre utilizzato del codice python per la gestione degli errori del server stesso, o per il reindirizzamento ad una pagina web differente da quella richiesta.

Per l'installazione (quale sono necessari appunto Apache Web Server e Python presenti nel sistema) dopo aver compilato la libreria o aver recuperato i pacchetti precompilati (presenti in quasi tutte le distribuzioni linux) è necessario aggiungere i comandi di configurazione al file httpd.conf di Apache:

```
LoadModule python_module libexec/mod_python.so
```

```
E
```

```
<Directory /some/directory/htdocs/test>
```

```
AddHandler mod_python .py
```

```
PythonHandler mptest
```

```
PythonDebug On
```

```
</Directory>
```

Dove `/some/directory/htdocs/test` va sostituito con la directory utilizzata dal sito web.

Per l'accesso lato server ai parametri inviati dal client è possibile utilizzare il comando:

```
val = req.form['initial']
```

la comunicazione server-client sarà descritta in dettaglio nel paragrafo "Comunicazione client/server"

3.5.3.5. *Numpy e Scipy*

Numpy è un'estensione per python che aggiunge il supporto per array multidimensionali e matrici anche molto grandi, insieme ad una libreria di funzioni matematiche di alto livello per operare su di essi. Il predecessore di NumPy, Numeric, fu originariamente creata da Jim Hugunin. NumPy è distribuita gratuitamente con licenza Open Source

La motivazione per l'utilizzo di NumPy sono le prestazioni. Poiché python è un linguaggio interpretato, operazioni complesse computazionalmente come quelle sugli array risultano essere molto più lente rispetto ad implementazioni, ad esempio in C, precompilate. Numpy consiste in una libreria di funzioni che contiene dei wrapper a funzioni scritte in C, migliorando notevolmente le prestazioni.

NumPy è pensato, inoltre come una alternativa free alla programmazione in MATLAB, in quanto utilizzano entrambi linguaggi interpretati e permettono di lavorare su matrici invece che su elementi scalari. Inoltre sono entrambi estendibili: Matlab tramite la tecnologia dei Toolbox e NumPy tramite le moltissime librerie esistenti per il linguaggio Python. Inoltre basata su NumPy è possibile utilizzare, sempre con licenza free open source, la libreria SciPy che aggiunge funzionalità per l'elaborazione scientifica avanzata (come l'analisi dei segnali numerici) al set base di NumPy. Infine sia Matlab che NumPy utilizzano la stessa libreria per l'elaborazione (LAPACK, Linear Algebra PACKage, anch'essa open source e con licenza free).

Per l'utilizzo della libreria è necessario incorporare nel codice Python la riga:

```
import numpy
```

La classe *array* di NumPy è implementata in maniera da semplificare il codice, eliminando molto spesso la necessità di loops per esempio la somma di due vettori con gli usuali costrutti del linguaggio

```
a = range(10000000)
```

```
b = range(10000000)
```

```
c = []
```

```
for i in range(len(a)):
```

```
    c.append(a[i] + b[i])
```

può essere facilmente implementata tramite Numpy con il codice:

```
import numpy as np
```

```
a = np.arange(10000000)
b = np.arange(10000000)
c = a + b
```

la prima versione risulta inoltre molto più lenta. Infatti, poiché il linguaggio utilizzato è a tipizzazione dinamica ad ogni loop l'interprete verifica il tipo di dato degli operandi a e b e seleziona l'opportuna variante per l'operatore "+". Nel secondo caso il controllo avviene una sola volta, e seleziona per l'operatore "+" tra due array di Numpy l'operazione di somma scritta in una funzione "C" compilata. La seconda versione risultò, perciò, molto più veloce.

3.5.4. Comunicazione client/server

Per l'elaborazione dei dati, creati dal client, da parte del server è necessario utilizzare una qualche forma di comunicazione client/server. Per questo lavoro è stato scelto il formato di scambio JSON elaborato sia da Javascript (tramite Dojo), sia da Python (con l'aiuto della libreria JsonLib)JSON

3.5.4.1. *JSON*

Acronimo di JavaScript Object Notation, JSON è un formato per la rappresentazione di dati (come XML) adatto per lo scambio dei dati in applicazioni client-server. È basato sulla sintassi degli oggetti del linguaggio Javascript (Standard ECMA-262 3a edizione dicembre 1999).

La semplicità di JSON ne ha decretato un rapido utilizzo specialmente nella programmazione in AJAX. JSON è, infatti, vantaggioso rispetto a XML per la maggiore facilità di lettura dei dati da parte dell'uomo, per una struttura più compatta del codice rispetto ad XML (che riduce, perciò la banda necessaria per la trasmissione dei dati), e per una somiglianza della sintassi a quella di Javascript (utilizzato lato client). Il suo uso tramite JavaScript, è particolarmente semplice grazie anche alle librerie (ad esempio i comandi *Dojo.toJson()* e *Dojo.fromJson()*). Questo fatto lo ha reso velocemente molto popolare a causa della diffusione della programmazione in JavaScript nel mondo del Web.

I tipi di dati supportati da questo formato sono gli stessi supportati nativamente da Javascript:

- booleani (true e false).
- interi, reali, virgola mobile.
- stringhe racchiuse da doppi apici (").
- array (sequenze ordinate di valori, separati da virgole e racchiusi in parentesi quadre []).
- array associativi (sequenze coppie chiave-valore separate da virgole racchiuse in parentesi graffe).
- null.

Un esempio di stringa Json è ad esempio:

```
{
  "users": [
    {
      "username": "user0",
      "indegree": 0.5,
      "reciprocity": 0.0,
      "outdegree": 0.0,
      "degree": 0.5
    },
    {
      "username": "user1",
      "indegree": 0.10000000000000001,
      "reciprocity": 0.0,
      "outdegree": 0.0,
      "degree": 0.10000000000000001
    },
    {
      "username": "user2",
      "indegree": 0.10000000000000001,
      "reciprocity": 0.0,
      "outdegree": 0.0,
      "degree": 0.10000000000000001
    },
    {
      "username": "user3",
      "indegree": 0.10000000000000001,
      "reciprocity": 0.0,
      "outdegree": 0.0,
      "degree": 0.10000000000000001
    }
  ],
  "group_reciprocity": 0.18181818181818182,
```



```
[1,1,0,2,1],  
[1,1,1,0,1],  
[0,0,0,0,0]],  
"users":["142","143","144","145","146"],  
"time":44,  
"meetingID":"2"}
```

I dati vengono poi serializzati in JSON tramite il comando `dojo.toJson()`:

```
toSend = dojo.toJson(Datas)
```

La stringa `toSend` sarà poi inserita come dati da spedire al server nel comando seguente

```
dojo.xhrPost({  
    url: "./serverside/matrix.py/get_stats",  
    handleAs: "text",  
    content: {  
        initial: toSend  
    },  
    load: function(response, args){  
        update_globalchart(dojo.fromJson(response))  
    },  
    error: function(response, args){  
        return response;  
    }  
});
```

il comando si occuperà di contattare il server e attenderà una risposta: in caso positivo, verrà eseguita la funzione `update_globalchart(dojo.fromJson(response))` (che mostrerà i dati elaborati tramite l'interfaccia grafica), in caso di errore si occuperà di gestire le eccezioni. Durante l'attesa il controllo ritornerà al programma principale senza bloccare l'elaborazione (per una descrizione dettagliata del comando `.xhrPost` si veda il capitolo Dojo).

Lato server una funzione Python (precisamente la funzione "get_stats" che si trova nel file `/serverside/matrix.py`) si occuperà di leggere i dati spediti dal client, di elaborarli e di spedire al client la risposta.

```
def get_stats(req):  
    val = req.form['initial']  
    datas = json.loads(val)  
    #codice di elaborazione mancante  
    return json.write(json_data, indent="    ")
```

in questo codice la funzione `get_stats` riceve come argomento I dati spediti dal client, legge il campo `initial` e lo registra nella variabile `val`.

La variabile `val` viene poi interpretata come JSON tramite la funzione `json.loads(val)`

L'elaborazione avrà come output l'oggetto (il codice python non è qui riportato):

```
json_data = {  
    "group_reciprocity": 0.0,  
    "centralization": 0.083333333333333329,  
    "users": [  
        {  
            "username": "user0",  
            "indegree": 0.0,  
            "reciprocity": 0.0,  
            "outdegree": 0.0,  
            "degree": 0.0  
        },  
        {  
            "username": "user1",  
            "indegree": 0.0,  
            "reciprocity": 0.0,  
            "outdegree": 0.25,  
            "degree": 0.25  
        },  
        {  
            "username": "user2",  
            "indegree": 0.0,  
            "reciprocity": 0.0,
```

```
        "outdegree": 0.25,  
        "degree": 0.25  
    },  
    {  
        "username": "user3",  
        "indegree": 0.25,  
        "reciprocity": 0.0,  
        "outdegree": 0.0,  
        "degree": 0.25  
    },  
    {  
        "username": "user4",  
        "indegree": 0.25,  
        "reciprocity": 0.0,  
        "outdegree": 0.0,  
        "degree": 0.25  
    }  
],  
    "density": 0.20000000000000001  
}
```

Tramite il comando

```
return json.write(json_data, indent="    ")
```

verrà infine spedita al client la risposta per l'elaborazione.

4. Fase di Design

4.1. Motivazioni

Attualmente la raccolta dati in reti sociali, viene solitamente fatta con strumenti non programmatici. Il principale strumento di raccolta delle informazioni è infatti il questionario (o intervista) nelle sue varie forme.

Esistono, però, software ormai consolidati ed efficienti per l'elaborazione dei dati (ad esempio Commetrix, <http://www.commetrix.de> con licenza commerciale, oppure Gephi, <http://gephi.org> con licenza OSS). Questi software, benché ormai utilizzati nella pratica di ricerca, hanno alcuni svantaggi, che non li rendono adatti al tipo rete considerato in questo lavoro.

Primo svantaggio sicuramente la difficoltà di utilizzo da parte di utenti senza una preparazione specifica in materia di analisi di rete, e teoria dei grafi. I software esistenti sono molto completi e performanti, ma l'enorme quantità di funzionalità si traduce in un'enorme quantità di menu, barre delle applicazioni, pulsanti e menu contestuali, rendendo difficoltoso l'addestramento all'utilizzo del software. Questo ne preclude, perciò, la possibilità di utilizzo a utenti "non addetti ai lavori", come ad esempio manager di azienda che potrebbero utilizzare questo tipo di software per monitorare la bontà relazioni esistenti tra i dipendenti dell'azienda, oppure altri operatori del mondo economico come addetti al marketing a cui questo tipo di analisi risulterebbe molto utili per migliorare le loro strategie di marketing, i quali potrebbero impiegare la social network analysis per studiare il rischio di diffusione di epidemie all'interno dei reparti ospedalieri rispetto ai rapporti tra infermieri e pazienti.

Molti software possiedono, poi, un'interfaccia grafica testuale in stile linea di comando, che richiede ulteriori conoscenze informatiche, la quale ne rende ancora più problematico l'uso.

Inoltre questo tipo di software è pensato per reti molto grandi, mentre la tipologia di studio richiesta in questo lavoro è quella delle piccole reti, per le quali, gli indici utilizzati per l'analisi e le problematiche di progettazione risultano molto diverse.

Lo strumento che si vuole realizzare si pone come obiettivo quello di utilizzare un metodo di raccolta dati più preciso e un sistema di elaborazione semplice ottimizzato per social network con pochi nodi, che sia facile da utilizzare da un'utenza non specializzata e non necessariamente appartenente al mondo accademico. La raccolta dati dovrà avvenire utilizzando un formato grafico che permetta la visualizzazione della rete anche in tempo reale e la registrazione degli eventi in maniera facile e chiara.

Benché il lavoro di questa tesi si concentra sull'analisi di riunioni, il software sviluppato dovrà essere facilmente adattabile ad altri tipi di situazione, che possa essere rappresentata con la stessa tipologia di rete.

4.2. Requisiti di base

Tramite interviste preliminari e Focus Group a cui hanno partecipato psicologi, sociologi, manager e esperti di Social Network Analysis sono stati ottenuti i seguenti requisiti, poi trasformati in requisiti per lo sviluppo:

8. Lo scopo del software che si vuole realizzare è l'analisi di social network dinamici in ben determinate situazioni: riunioni, meeting, discussioni pubbliche. Il software si dovrà occupare, infatti, di rappresentare la disposizione dei partecipanti nella stanza nella quale avviene l'incontro e l'interazione tra le persone partecipanti. Dovrà essere poi possibile effettuare un'analisi sia in tempo reale in preciso istante della riunione, sia a posteriori tramite strumenti di analisi adeguati messi a disposizione dal software stesso. Il software potrà essere infine utilizzato sia per monitorare meeting reali, sia per la simulazione. Data la natura del social network da analizzare si è progettata l'applicazione per l'utilizzo con piccoli gruppi (dalle 5 alle 40 persone). Si è supposto che il software possa essere gestito durante il meeting da una sola persona, come ad esempio il coordinatore della riunione, oppure un ricercatore, che si impegnerà a registrare gli eventi che avverranno nel corso della discussione. Il software infine deve fornire informazioni dettagliate per l'analisi della rete, però deve essere facilmente utilizzabile e interpretabile da utilizzatori non esperti né in Social Network Analysis né in informatica, con interessi anche molto diversi da quelli di ricerca (ad esempio manager, esperti di risorse umane, analisti di marketing, ecc.), costituendo un ausilio per l'analisi applicata non solo ai classici scenari di ricerca scientifico-psicologici, ma anche a scenari economico-sociali presenti nelle aziende.
- Il software deve essere facilmente adattabile a situazioni diverse che richiedano l'analisi di reti di piccole dimensioni, formate da persone e non; un primo sviluppo si dovrà concentrare sull'analisi delle reti di riunioni, e, per testare l'effettiva adattabilità a varie situazioni.
- All'interno del software dovrà essere chiaramente visualizzata la rete sulla quale si va ad operare e le azioni compiute sulla rete reale dovranno essere rappresentate da altrettante azioni sulla rete virtuale legate tra loro tramite legami semantici: ad esempio l'evento "A parla con B" sarà rappresentata tramite un click sul nodo A e un secondo sul nodo B ad indicare i soggetti coinvolti nell'azione e la direzione dell'evento. Anche i nodi, soggetti, o entità della rete reale dovranno avere un chiaro corrispettivo nella rete virtuale. Nel caso in esame della rete di una riunione i nodi saranno rappresentati da un'icona somigliante al partecipante o da una sua foto. Dovrà essere possibile avere una traccia visiva dei nodi attivi e non attivi all'interno della rete. Nel caso delle riunioni per creare una situazione più realistica, dovrà essere possibile selezionare un tavolo e una texture del pavimento che somiglino all'arredamento reale della stanza nella quale avviene l'incontro.
- La fase di setup, nella quale verrà creata la rete dovrà essere veloce, ma avere la possibilità di essere perfezionata per inserire le informazioni sugli utenti. Durante la fase di "setup" si potranno trascinare all'interno dell'area grafica che rappresentano il network, tramite *drag and drop*, le icone che andranno a rappresentare la persona
- La raccolta dati dovrà avvenire registrando gli eventi richiesti dall'applicazione per il calcolo delle statistiche tramite le azioni definite: per il caso delle conversazioni saranno registrate le azioni "parla con", entra ed esci dalla stanza, "Parla a tutti" e cambia posto.

- L'analisi sulla rete dovrà poter essere effettuata in tempo reale, e successivamente, off-line utilizzando degli indici (descritti nel capitolo "Misure") che siano validi descrittori della tipologia di rete utilizzata e siano presentati all'utente in modo da permettere una facile interpretazione.
- L'interfaccia dovrà essere sviluppata in modo da essere adatta all'utilizzo con un dispositivo con schermo touch: si dovranno eliminare perciò menu molto profondi e azioni da eseguire con il tasto destro del mouse; inoltre le dimensioni degli elementi grafici dovranno essere tali da permettere un agevole utilizzo tramite pennino o dito.

4.3. Possibili ambiti di utilizzo

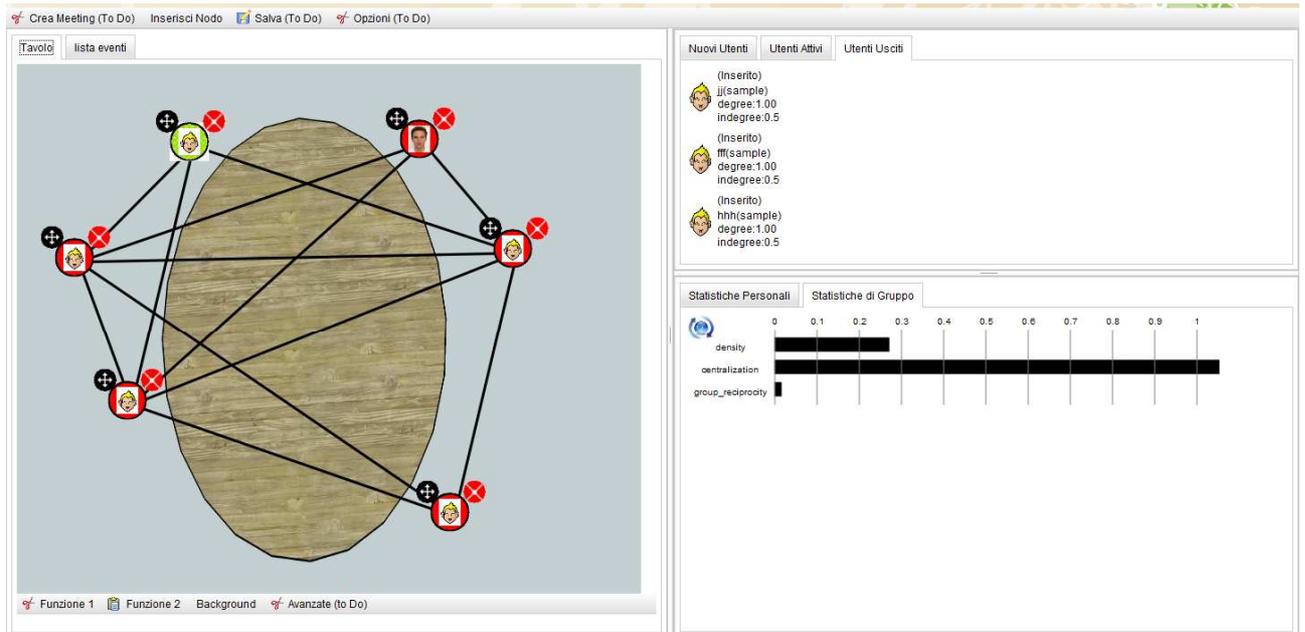
Di seguito sono riportati alcuni esempi dei tipici ambiti in cui un software di questo tipo, può rivelarsi utile.

Una prima categoria di utenza a cui si è ritenuto possa essere utile un software di questo genere è quello del manager d'azienda, o di un addetto alle risorse umane. Si ponga il caso, ad esempio di un gruppo di persone facenti parte del team di sviluppo di un prodotto o servizio, nel quale esiste una gerarchia formale: il capo del progetto, i vari responsabili, i tecnici. In un gruppo di questo tipo, durante lo sviluppo del progetto, si potrebbe formare una rete di relazioni ben diversa da quella definita. Ad esempio potrebbe esserci un tecnico non responsabile o capo progetto addetto allo sviluppo di una parte principale, al cui, necessariamente, la maggior parte degli altri soggetti coinvolti si dovranno rivolgere, diventando un nodo centrale della rete. Viceversa si potrebbe creare la situazione, nella quale un responsabile inesperto non riesca nel coordinamento delle parti, creando una situazione nel quale il soggetto risulterà poco connesso con gli altri nodi della rete. Utilizzando un software come quello sviluppato in questo lavoro, è possibile tramite un'analisi precisa delle connessioni rintracciare queste situazioni per poter modificare le gerarchie formali per renderle più simili a quelle effettive all'interno del gruppo di lavoro.

Un secondo esempio legato al team di sviluppo è quello della riunione, o focus group, per la progettazione o il controllo dello stato di un progetto. In questi casi, soprattutto nelle fasi iniziali, dove i partecipanti non hanno spesso avuto rapporti lavorativi precedenti o, nel caso del design collaborativo di un prodotto, i partecipanti sono stati scelti tra i vari possibili utilizzatori e non si conoscono completamente, si possono formare delle relazioni legate al carattere delle persone: ci sarà qualcuno con un carattere più forte, che tenderà a diventare un nodo principale della rete, qualcun altro che tenderà a parlare poco diventando un nodo più periferico. Durante questo tipo di riunioni il compito del coordinatore della riunione è quello moderare la discussione facendo in modo di far esprimere le proprie idee in egual misura a tutti i soggetti coinvolti, evitando così, di considerare un'idea migliore delle altre solo perché è stata espressa con più forza, oppure scartare una buona idea perché non condivisa da alcuni soggetti nodi focali della rete. Tenendo traccia delle conversazioni avvenute tra i partecipanti e registrandole nel software è possibile conoscere in tempo reale chi sono i nodi centrali della rete, e chi i punti isolati, facilitando il lavoro di coordinamento. In un'analisi a posteriori è poi possibile valutare quali sono le persone risultate più utili, e quali, invece, tendono ad attirare l'attenzione oppure hanno paura di esprimere le proprie idee, potendo così migliorare il gruppo di lavoro in futuro.

5. Fase di Sviluppo

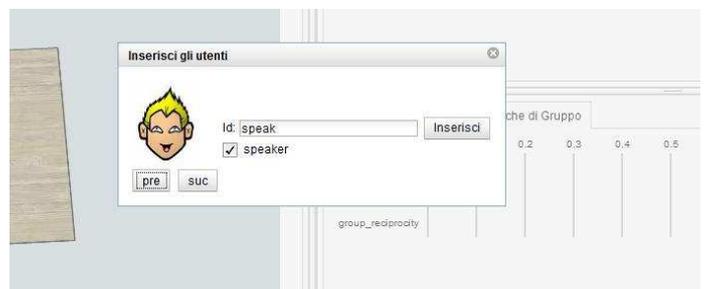
5.1. Versione alpha



La prima versione preliminare del software, denominata “*alpha*”, è stata sviluppata per studiare in dettaglio i principali requisiti dell’interfaccia da inserire, poi, nella versione finale.

L’applicazione è costituita da una sola schermata divisa in due parti: una parte sinistra contenente la rappresentazione della rete, e una destra contenente i dati sulla rete e la lista di utenti.

Per l’inserimento dei soggetti è necessario far click sul pulsante della barra in alto “inserisci nodo”, nel quale apparirà una schermata dove andrà inserito il nome dell’utente e scelta la foto del soggetto. Cliccando su inserisci (o premendo il tasto “invio” se nel sistema è presente la tastiera) l’icona del soggetto verrà inserito al centro dell’area di lavoro e, tramite i comandi posti al di sopra dell’icona sarà possibile spostarlo nella posizione desiderata.



Tramite i click sulle icone è possibile registrare gli eventi:

A parla con B: per registrare questo evento occorre fare click sull'icona del soggetto A e poi un secondo click sull'icona del soggetto B.

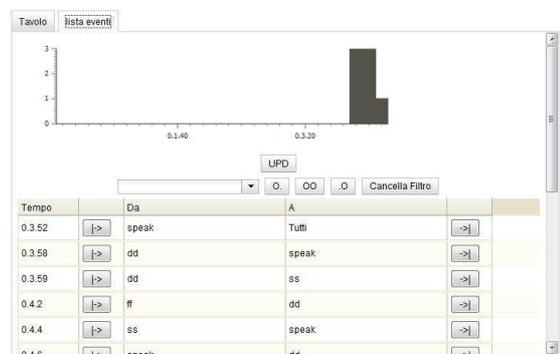
Parla a Tutti: per registrare questo evento basta effettuare un doppio click sull'icona del soggetto considerato.



A Parla con un gruppo di utenti: occorre fare click sul soggetto sorgente a e cliccare sul pulsante che appare al di sotto dell'icona, una volta selezionato occorre selezionare i soggetti destinatari e poi cliccare di nuovo sull'icona del soggetto sorgente.

Ad intervalli di tempo prestabiliti il client dell'applicazione manda gli eventi ad un server che si occupa del calcolo della matrice e dell'invio degli indici calcolati.

Cliccano sul tab "lista eventi" in alto è possibile visualizzare i dati cronologici sulla sessione: una tabella contenente la lista dettagliata degli eventi avvenuti nella rete e un istogramma che rappresenta il numero di eventi avvenuti in intervalli di tempo di dieci secondi a partire dall'inizio della sessione. È possibile inoltre filtrare tramite vari criteri i dati presenti.



Trattandosi di una prima versione di test l'applicazione prevedeva l'uso di una sola sessione cancellata ogni volta venga effettuato l'accesso.

Nella parte destra è possibile distinguere due sezione.

La sezione superiore contiene la lista di tutti gli utenti: tramite i tre "tab" è possibile scorrere la lista completa degli utenti, la lista degli utenti attivi e la lista degli utenti usciti dalla discussione. Per modificare lo stato dell'utente è necessario cliccare sul pulsante X presente vicino all'icona dei soggetti.

La sezione inferiore contiene le statistiche visualizzate tramite un istogramma divise in "statistiche globali" e "statistiche personali".

Da un primo test sull'uso di questa prima versione sono risultati alcuni problemi.

La parte destra è risultata di complicata fruizione. La parte superiore, senza ulteriori informazioni risulta confusa e con informazioni ridondanti mentre la parte inferiore contiene tante informazione che necessitano di una visualizzazione più immediata senza la necessità di ricercare tra le schermate l'informazione richiesta. Inoltre non c'è distinzione tra l'analisi dei dati e la raccolta, che avviene utilizzando la stessa schermata.

La schermata "lista eventi" si è, infine, rivelata essere poco utile per l'analisi.

5.2. Versione beta

5.2.1. I moduli

Per mimare al meglio le varie fasi di una riunione è stata pensata per l'applicazione una struttura a moduli: "CREATE/LOAD", "ADD PARTICIPANT", "DRAW CONVERSATION", "ANALYSIS".

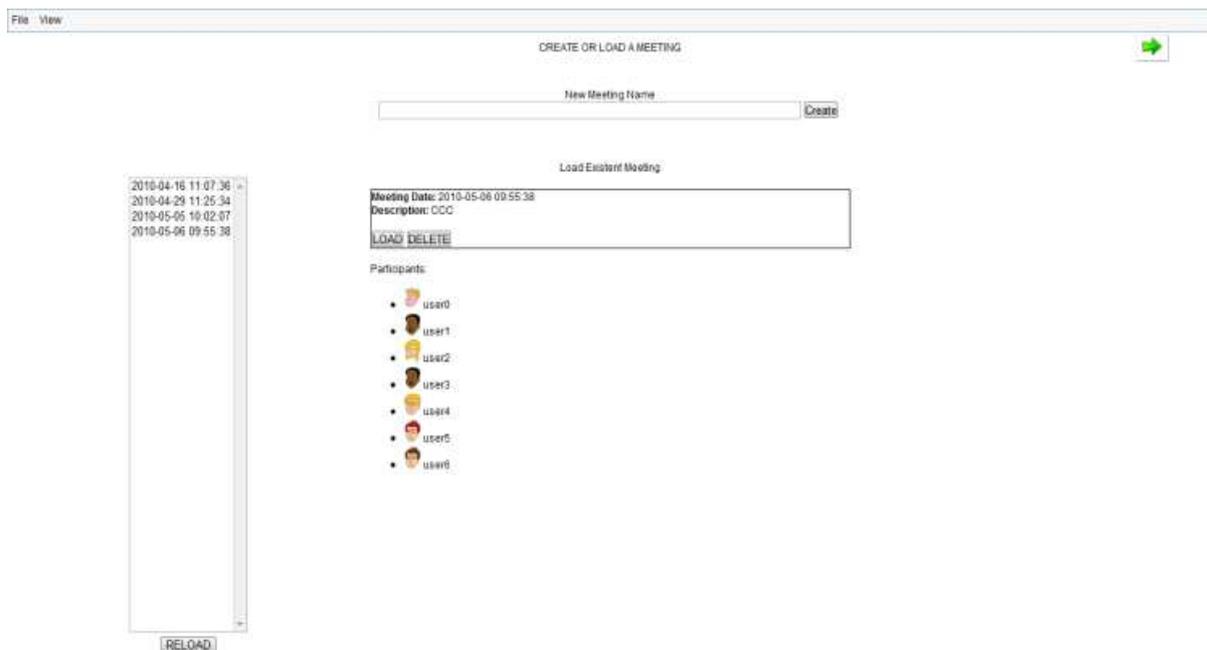
Il menu e la navigazione tra i moduli



Il software presenta in alto un classico menu con funzioni di base di gestione. Esso è contestuale e varia a seconda del modulo utilizzato, in particolare nel modulo "SETUP" è presente, oltre al menu "FILE" e "VIEW" presenti in tutti gli altri, il menu "EDIT" che permette di aggiungere utenti e sfondi all'area di lavoro.

La navigazione tra i moduli può invece avvenire in due modi. Tramite il menu "VIEW" è possibile, se lo stato dell'applicazione lo permette, saltare un modulo ad un qualsiasi altro. Per essere invece guidati nelle operazioni da compiere in sequenza logica è stata inserita subito al di sotto del menu una barra che mostra in centro il nome del modulo selezionato un freccia a destra ed una a sinistra: tramite la freccia destra è possibile spostarsi nel modulo successivo della sequenza logica (dopo aver effettuato tutte le operazioni richieste dal modulo corrente); selezionando la freccia di sinistra è possibile, invece tornare al passo logico precedente.

5.2.1.1. *Modulo CREATE/LOAD*



Questo primo modulo vuole rappresentare la creazione della meeting. È possibile creare un record, nel database che contenente i dati dell'applicazione, che rappresenta il meeting, inserendone la descrizione e la data di creazione. Da questa schermata è possibile inoltre caricare, ed eliminare, un meeting precedentemente creato.

5.2.1.2. Modulo ADD PARTICIPANT



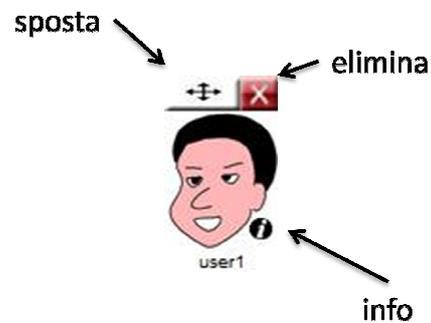
Il secondo modulo serve ad inserire all'interno del meeting virtuale tutte le informazioni necessarie alla preparazione del meeting. È qui possibile definire le persone che partecipano al meeting le loro generalità, e il layout della stanza. La schermata è divisa in due parti.

La sezione di sinistra contiene la rappresentazione della stanza vista dall'alto della riunione. Questa rappresentazione consta di tre layer: il layer di base contenente la texture del pavimento della stanza, un secondo layer contenente una texture del tavolo e un terzo layer contenente le icone dei partecipanti al meeting.

Nella parte destra è invece inserita una toolbox, creata tramite il componente *Accordion* di Dojo, divisa in tre sottosezioni corrispondenti agli altrettanti layer della parte sinistra. La prima sezione denominata "People" contiene le icone presenti che andranno a rappresentare i partecipanti al meeting. Una volta individuata l'icona è possibile, tramite Drag and Drop trascinarla all'interno

dell'area di lavoro (parte sinistra). Una volta inserita l'icona, viene data la possibilità all'utilizzatore di inserire l'ID dell'utente tramite inputbox sottostante all'icona inserita, oppure di assegnarne uno automatico.

All'interno dell'icona inserita esistono alcune aree interattive:



Tramite la barra di ricerca è possibile anche selezionare un utente già inserito nel database perché partecipante ad un altro meeting. Basterà immettere una stringa contenuta nell'ID dell'utente e premere il pulsante *Filter* il software si occuperà di reperire gli utenti ricercati e di mostrare l'icona e l'ID al posto della lista delle icone disponibili. Con le stesse modalità dell'inserimento di un nuovo utente è possibile trascinare l'utente all'interno dell'area di lavoro.

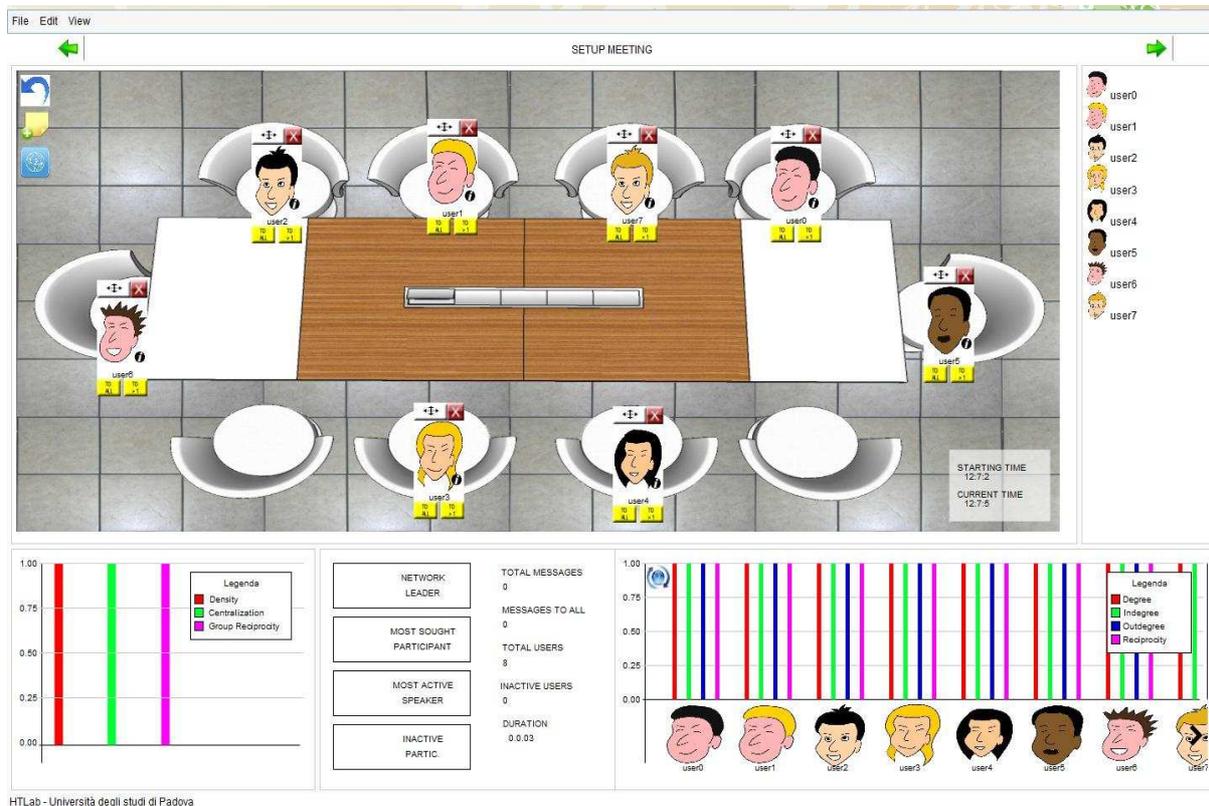
Nella seconda sezione (*Setting*) sono presenti le miniature dei vari tavoli che è possibile inserire sempre tramite drag and Drop nella parte sinistra.

L'ultima sezione (*Background*) contiene, invece, le varie texture del pavimento.

Il software viene fornito di una collezione di texture e icone dei partecipanti di base, alle quali è comunque possibile aggiungerne delle altre tramite i menu del software stesso. Le texture verranno inserite con la stessa risoluzione dell'immagine di partenza, mentre le icone verranno ridimensionate alla risoluzione di 100X80 pixel per ridurre lo spazio necessario su server e ridurre il tempo di caricamento dell'immagine nell'applicazione. È possibile così inserire una foto reale dei partecipanti aumentando la facilità di utilizzo del software stesso. Si ipotizza per una prossima release la possibilità di inserire immagini direttamente tramite una webcam collegata al dispositivo dal quale si sta utilizzando l'applicazione, velocizzando così il processo di preparazione del meeting, funzionalità però non supportata dai browser al momento.

Il pulsante "START MEETING" permette di avviare l'analisi della rete appena inserita passando alla visuale successiva.

5.2.1.3. Modulo DRAW CONVERSATION



In questo modulo è possibile interagire con la stanza virtuale registrando i vari eventi.

L'interfaccia grafica è divisa anche qui in due parti: l'area superiore può essere utilizzata per l'inserimento degli eventi, l'area inferiore è adibita all'analisi della rete.

Nella parte superiore è possibile trovare un'area di lavoro molto simile a quello presente nel modulo precedente, ma con alcune importanti differenze. In questa vista è possibile infatti compiere operazioni differenti sugli avatar dei partecipanti. Sono inoltre aggiunti due pulsanti per la registrazione di particolari eventi.

Gli eventi considerati nella pagina sono di tre tipi: SPEAK e ENTER/EXIT e MOVE

Evento SPEAK

Questo tipo di evento rappresenta l'atto che compie un partecipante di parlare ad uno o più degli altri e viene registrato nel database differentemente a seconda dei casi, viene cioè assegnato il tag SPEAKTO se l'interlocutore è univoco, il tag SPEAKTOMANY se l'interlocutore è più di un ma non l'intera assemblea oppure il tag SPEAKTOALL se invece l'utente si rivolge a tutti. Ai vari eventi viene inoltre associato un "timestamp".

L'evento di tipo SPEAKTO è inseribile cliccando prima sull'avatar del soggetto che parla (all'area verrà aggiunto un bordo giallo), e successivamente su quella del soggetto che ascolta (all'area del soggetto

parlante verrà cancellato il bordo). Con il secondo click viene, infine tracciata una linea nera tra il soggetto “sorgente” e quello “ricevitore” che andrà a scomparire entro pochi secondi.

L’evento SPEAKTOMANY può essere invece registrato cliccando sul pulsante “To>1” (presente sotto l’icona di ogni avatar) presente vicino all’avatar “sorgente”; in questo caso l’icona avrà un bordo blu. Si dovranno poi selezionare le varie icone dei soggetti “ricevitori”, che verranno bordate di giallo. Una volta identificati tutti i “ricevitori” si farà click sul pulsanti “To>1” di partenza e l’evento verrà registrato. In questo caso verranno tracciate delle linee nere tra l’avatar sorgente e i vari ricevitori che scompariranno nel giro di pochi secondi.

L’evento SPEAKTOALL, infine è attivabile facendo un doppio click sull’icona del soggetto “sorgente” oppure cliccando una volta sul pulsante “To All” relativo al all’avatar. In questo caso verranno disegnate delle linee nere dal soggetto sorgente a tutti gli altri partecipanti che scompariranno dopo pochi secondi.

Eventi ENTER/EXIT

Questi due eventi si riferiscono alla presenza o meno di un determinato soggetto in un determinato momento della riunione oppure no. All’inizio della riunione il sistema si occupa di inserire un evento ENTER (con timestamp 0) per ogni soggetto inserito, che ogni soggetto sia presente all’inizio della riunione. Se così non fosse è possibile, prima di registrare qualsiasi altro evento, inserire un evento EXIT per ogni persona non presente alla discussione alla partenza.

Per inserire l’evento EXIT è necessario cliccare sul pulsante  relativo alla persona interessata. Se per esempio una determinata persona si assenta dalla riunione qualche minuto per una telefonata, sarà possibile cliccare sul pulsante e l’icona relativa verrà nascosta.

La barra laterale destra presente nella sezione superiore mostra, invece, la lista degli utenti presenti mostrandone l’icona e l’ID. Questo sarà presentato in nero se la persona è presente all’incontro, in blu se al momento è assente dalla stanza: da qui tramite Drag-and-Drop è possibile reinserire l’icona trascinando il nominativo richiesto all’interno dell’area di lavoro. L’icona ritornerà visibile nello stesso punto dove era precedentemente e nella barra laterale verrà ripresentata con testo nero. Tramite questa operazione verrà inoltre registrato un evento di tipo ENTER.

Evento MOVE

Questo evento corrisponde all’azione “cambiare posto” da parte di un soggetto partecipante. Per registrare l’evento MOVE è necessario cliccare sull’icona  vicina all’immagine del soggetto e trascinare così l’avatar nella nuova posizione.

Note



Tramite il pulsante *Addnote* può essere utilizzato per segnalare al software alcune istanti particolari (che verranno chiamati “note”). Tramite la schermata *Analysis* potranno poi essere associati ad appunti testuali che potranno aiutare l’analisi della discussione.

UnDo



Il software è fornito di un sistema di Undo. Cliccando sul pulsante *Undo* è possibile eliminare l’ultimo evento registrato. Per mostrare che l’eliminazione sia andata a buon fine il feedback ottenuto sarà differente a seconda del tipo di evento eliminato: per gli eventi di tipo SPEAK verranno tracciate delle linee rosse, che scompariranno gradatamente dopo pochi secondi, tra la sorgente le altre persone interessate; per gli eventi ENTER e EXIT, verrà rispettivamente nascosta o visualizzata l’icona dell’utente interessato all’interno dell’area di lavoro.

Visualizzazione delle connessioni della rete



Utilizzando il pulsante *ShowArcs* è possibile in qualsiasi momento visualizzare lo stato delle connessioni tra i partecipanti della rete. Cliccando sul pulsante tra i vari partecipanti verranno disegnati i vari archi con uno spessore proporzionale al peso che l’arco corrispondente ha all’interno del network.

Infine nella parte bassa dell’area di lavoro è presente l’indicazione dell’orario attuale (utile in quanto l’applicazione è pensata per l’utilizzo in modalità fullscreen del browser e l’orologio di sistema non è visualizzato) e dell’orario di inizio della sessione (nel caso di meeting salvati precedentemente e ricaricati sarà visualizzata l’ora del caricamento del meeting).

La parte inferiore di questo modulo contiene invece tutte le informazione necessarie per l’analisi in tempo reale dei dati.

La schermata è divisa in tre parti: a sinistra le informazioni sugli indici globali della rete, a destra le informazioni sugli indici legati ai singoli utenti, e una parte centrale con delle statistiche di vario tipo.

Indici globali

Data la natura della rete considerata (rete dinamica con pochi nodi) sono stati ritenuti interessanti per lo studio tre indici globali sullo stato della rete denominati *Coesion* (corrispondente alla Densità), *Involvement* (Centralizzazione) e *Democracy* (Group Reciprocity). I tre indici vengono calcolati utilizzando una matrice per la rappresentazione della rete dal server ogni 10 secondi e visualizzati tramite istogramma. Per facilità di gestione e visualizzazione gli indici scelti vengono normalizzati, il loro valore potrà perciò oscillare tra 0 e 1. Il valori dei vari indici così calcolati vengono visualizza come le barre di un istogramma: Barra rossa per *Coesion*, barra verde per *Involvement* e barra fuxia per *Democracy*. Nel riquadro è presente anche una legenda che distingue i vari indici. Cliccando

sopra il colore desiderato è poi possibile visualizzare una descrizione operativa per l'interpretazione del valore fornito da tale indice.

Indici locali

Oltre agli indici globali vengono calcolati quattro indici legati, invece, ai singoli nodi della rete, cioè ai singoli utenti: *Leadership*(Grado), *In*(Grado in entrata), *Out*(Grado in uscita), *Reciprocity*(Reciprocità). In questo caso gli indici vengono sempre visualizzati tramite istogramma e associati ad un colore: Rosso per *Leadership*, Verde per *In*, Blu per *Out*, Fuxia per *Reciprocity*. Nel grafico sono inoltre state inseriti gli avatar e gli ID dei partecipante sull'asse X dell'grafico e, per ogni utente, vengono visualizzati raggruppati al di sopra dell'icona corrispondente i quattro indici associati. Anche in questo caso è presente una legenda e cliccando sopra sopra il colore desiderato è possibile leggere una descrizione operativa per l'interpretazione dell'indice. In questo riquadro è infine presente un pulsante nel quale è possibile cliccare per aggiornare il calcolo degli indici senza la necessità di dover attendere i 10 secondi.

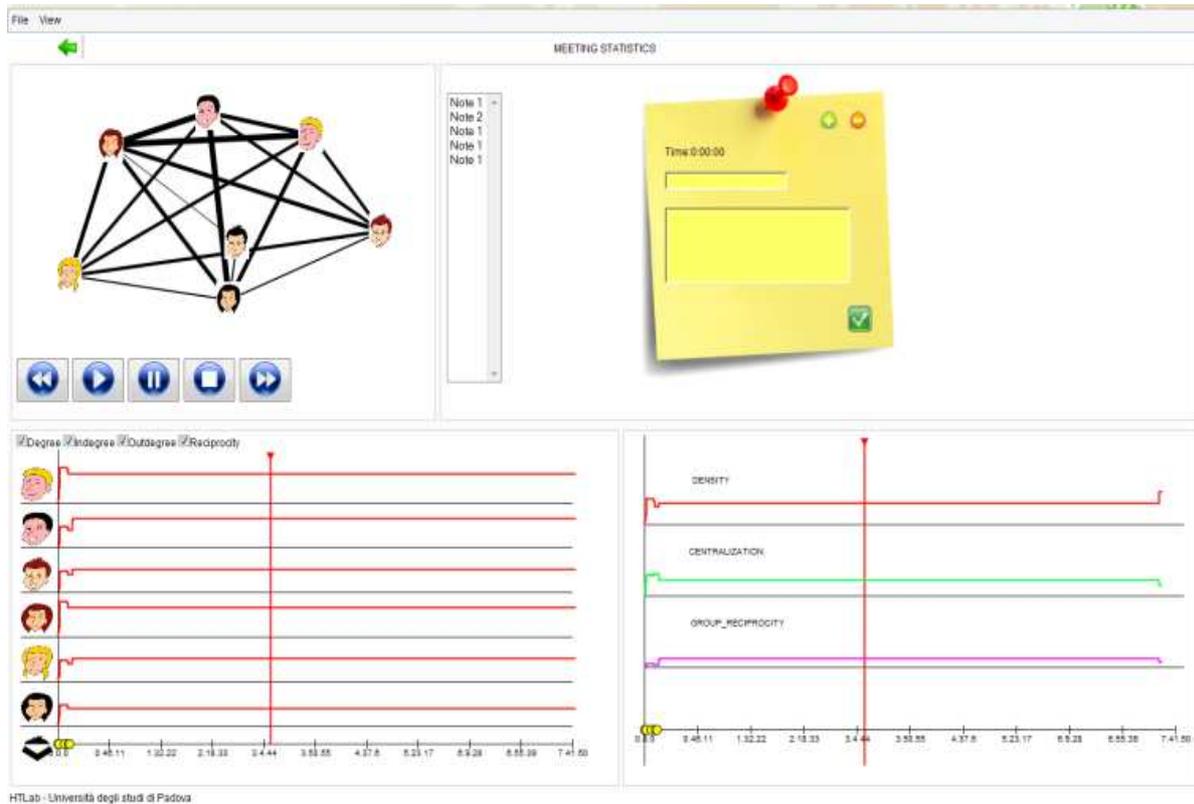
Riquadro centrale

Si è ritenuto utile per una migliore interpretazione dei risultati ottenuti inserire delle informazioni aggiuntive nel riquadro centrale. In questa sezione è possibile vedere 4 avatar corrispondenti ai quattro indici locali scelti. In particolare al variare della rete vengono visualizzati:

- **Network leader:** questo riquadro mostra il partecipante al meeting più attivo, ricavato scegliendo il nodo che al momento dell'ultima analisi della matrice risulta avere il valore di "Degree" più alto.
- **Most sought participant:** questo riquadro mostra il partecipante al quale più persone hanno parlato (il più ricercato), ricavato scegliendo il partecipante che risulta avere il valore più alto di "Indegree"
- **Most active speaker:** qui viene mostrato l'avatar del partecipante che ha parlato di più, ricavato scegliendo l'utente con maggior "Outdegree"
- **Inactive participants:** qui vengono mostrati gli utenti che hanno interagito di meno con il gruppo e che possiedono quindi il valore più basso di "reciprocity"

In questa sezione sono poi visualizzate altre informazioni ausiliarie: il numero totale di messaggi scambiati, il numero di messaggi scambiati del tipo "TOALL", il numero totale di utenti, il numero di utenti intattivi (con "reciprocity minima"), e infine la durata del meeting momento del primo click al pulsante start meeting nel modulo "ADD PARTICIPANT" (nel caso di meeting caricato verrà considerato come tempo di base il timestamp dell'ultimo evento inserito nelle sessioni precedenti).

5.2.1.4. Modulo "ANALYSIS"



In questo modulo avviene l'analisi off-line del social network associato al meeting. La schermata si divide in 4 parti: il video, il gestore delle note e le due Timeline

Il video

Nel riquadro in alto a sinistra è visualizzata il grafo del social network dinamico creato durante la fase precedente di discussione vera è proprio. Utilizzando gli usuali pulsanti "play", "pause", "stop", ecc. è possibile analizzare nel tempo la rete e vederne lo stato nei vari istanti. Il grafo qui visualizzato presenta, per una maggiore facilità di lettura gli avatar degli utenti partecipanti nel rispettivo nodo e un diverso spessore dell'arco congiungente due nodi variabile a seconda dell'importanza del nodo nel determinato istante di tempo. Lo spessore del nodo varia secondo un algoritmo molto semplice:

bisogna innanzitutto definire lo spessore massimo che si vuole dare all'arco, definito dalla variabile $max_thickness$, e calcolare il peso massimo che nella rete possono avere gli archi, calcolando i pesi negli istanti finali e prendendo il peso massimo tra i vari archi (poiché, per la natura della rete il peso varia di un numero positivo ad ogni evento il peso finale corrisponderà anche a quello massimo). Per ogni istante è quindi possibile definire lo spessore utilizzando la formula

$$arc_{thickness} = \frac{max_{thickness} * arc_{weight}}{max_{weight}}$$

Dove max_weight è il peso massimo raggiunto dai nodi e arc_weight il peso del nodo nell'istante considerato.

Timeline sinistra

In questa sezione vengono visualizzati gli andamenti nel tempo degli indici locali scelti relativi agli utenti. Tali indici sono rappresentati a fianco dell'avatar dell'utente associato in un piano cartesiano con in asse x il tempo e in asse y il valore normalizzato nell'istante. Nella parte alta sono presenti 4 checkbox che permettono di visualizzare o nascondere gli indici desiderati nel grafico. Nella finestra è inoltre visualizzata una barra verticale posizionata sul valore della x che indica l'istante visualizzato nella sezione video e cliccando in qualsiasi area dello schermo è possibile spostare la barra, cambiando nel contempo l'istante visualizzato nel video. Inoltre nell'asseX, in basso nel riquadro, sono visualizzati dei segnaposto che mostrano gli istanti di tempo ai quali sono associate delle note. Cliccando su tali segnaposto verrà selezionata la nota associata e nel video l'istante di tempo considerato.

Timeline destra

In questa sezione vengono visualizzati, con le stesse modalità della timeline di sinistra (piani cartesiani tempo x valore) i tre indici globali selezionati. Anche qui è visualizzata la barra verticale del tempo e i segnaposto per le note.

Note

In questa sezione è possibile gestire le note. Tutte le note inserite sono elencate sulla sinistra del riquadro, cliccando sul titolo, tutte le informazioni relative (istante associati, titolo e descrizione) sono visibili nel form sulla destra. Da qui inoltre è possibile eliminare ed aggiungere le note: bisogna selezionare l'istante di tempo interessato (tramite le timeline o la pulsantiera) e premere il pulsante add. Una volta inserita la nuova nota apparirà nell'elenco e sarà possibile selezionarla e modificarla.

5.2.2. Network Analysis e raccolta dati

A differenza della versione "alpha" la creazione della matrice di adiacenza del grafo viene calcolata lato client, in quanto la matrice può essere facilmente calcolata in maniera incrementale man mano che si aggiungo eventi nella rete, e questo calcolo influisce molto poco sulle prestazioni del client ed riduce il carico di lavoro del server. Una volta inviata, l'ultima matrice ricevuta dal server viene salvata nel database contenente i dati del meeting. Gli indici calcolati sono quelli descritti nel capitolo misure:

Centrality Degree, Centrality Indegree, Centrality Outdegree (I tre indici per la centralità), Reciprocity (reciprocità dei vari soggetti), Group Reciprocity (reciprocità di gruppo), Centralization (centralizzazione), e Density (densità).

La comunicazione tra client e server avviene tramite stringhe formattate in JSON.

Per visualizzazione del video nel modulo "ANALYSIS" il server si occupa di recuperare gli eventi della sessione e di fornire al client tre stringhe di dati la prima contenente i dati video, la seconda il dati per il rendering degli indici globali e la terza per gli indici locali.

In particolare il flusso video è creato dividendo gli eventi in un numero di frame dipendente dalla risoluzione dello schermo del client e dalla durata del meeting (ad esempio se la risoluzione è 800x600 e la durata del meeting è di 200 minuti il flusso video sarà di $800/2=400$ frames, e ogni frame sarà della durata di 30 secondi). Vengono poi per ogni frame selezionati gli eventi con *timestamp* compreso nell'intervallo di tempo del frame e vengono ricavati quale sono i nodi visibili nella rete e la loro posizione in quell'intervallo e quali sono gli archi da disegnare e con quale peso (ad esempio se A parla con B per 3 volte e B parla con A 2 volte nell'intervallo, l'arco (A,B) verrà incrementato di 5 unità in peso). I dati così interpolati verranno poi inseriti in un oggetto *video* e inviati con codifica JSON al client.

Un estratto dei dati spediti dal server:

```
{
  "TimeGrain": 1.5825,
  "users": [
    {
      "username": "user0",
      "X": 301,
      "Y": 68,
      "icon": "01.jpg"
    },
    {
      "username": "user1",
      "X": 186,
```

```
"Y": 57,  
"icon": "02.jpg"  
}
```

/Altro codice/

```
],  
"ResolutionY": 320,  
"ResolutionX": 480,  
"MaxArcsValue": 4,  
"frames": [  
  [  
    "0.v",  
    "1.v",  
    "2.v",  
    "3.v",  
    "4.v",  
    "5.v",  
    "6.v",  
    "7.v"  
  ],  
  [],  
  [],  
  [  
    "1.0.1",  
    "1.2.1",  
    "1.3.1",  
    "1.4.1",  
    "1.5.1",  
    "1.6.1",  
    "1.7.1"  
  ],  
],
```

```
    [],  
    [],  
  
/Altro codice/
```

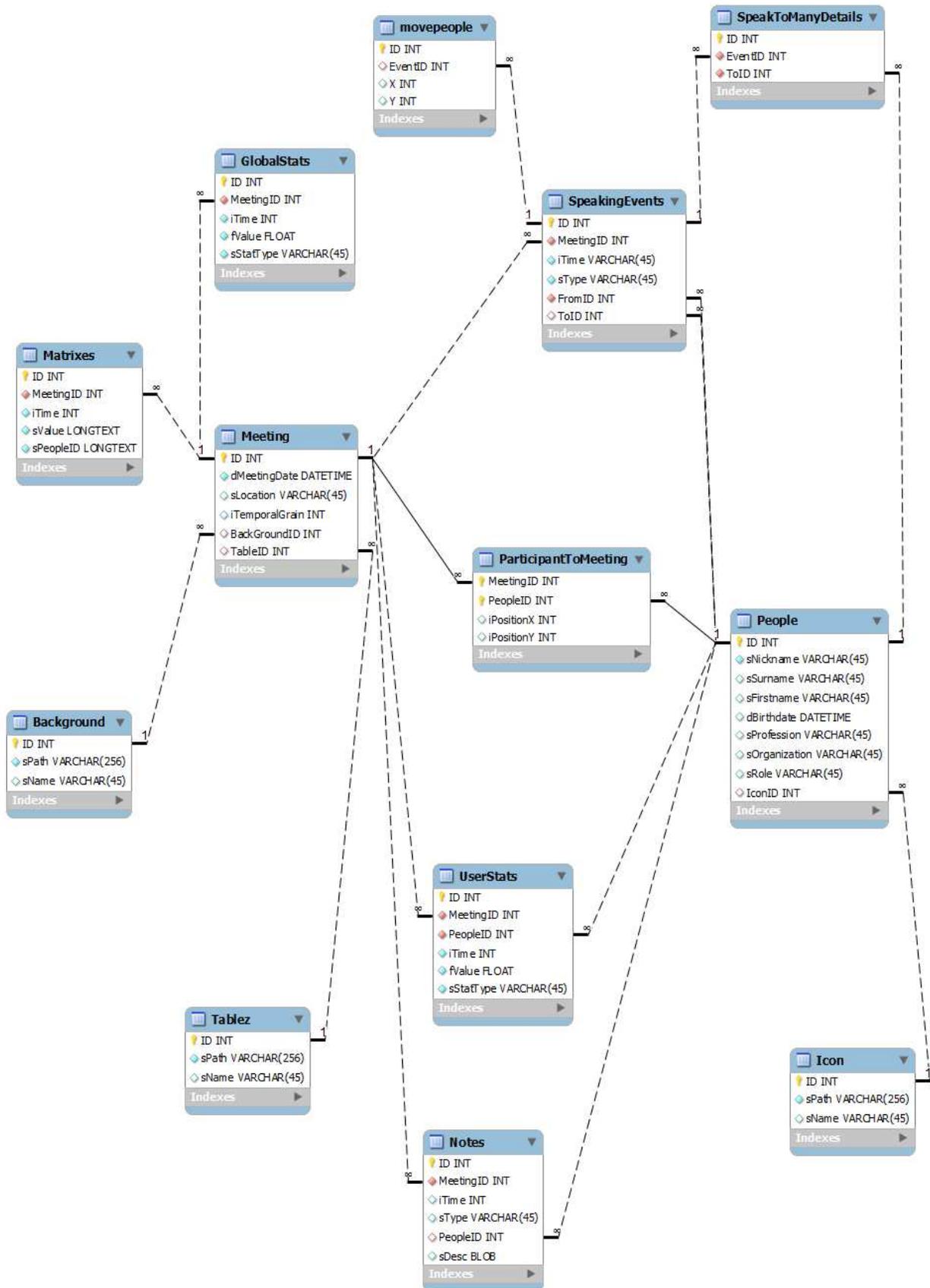
```
    ],  
    "MaxTime": 633,  
    "MeetingID": "7"  
}
```

Seguendo la stessa divisione in frames vengono poi calcolati i vari indici (globali e personali) scegliendo i valori (che sono stati, durante la fase di analisi, registrati nel database insieme agli eventi) con timestamp compreso nell'intervallo ed eseguendo tra loro una media.

Un esempio di codice:

```
{  
  "MeetingID": "7",  
  "DENSITY": "M20 100L118 100L118 69L158 69L158 99L236 99L236 72L640 72L640  
58L644 58",  
  "FPS": 1.0209677419354839,  
  "GROUP_RECIPROCITY": "M20 260L118 260L118 248L236 248L236 250L471 250L471  
245L640 245L640 250L644 250",  
  "MaxTime": 633,  
  "CENTRALIZATION": "M20 180L118 180L118 163L158 163L158 172L236 172L236 158L640  
158L640 165L644 165",  
  "Resolution": 620,  
  "users": [  
    {  
      "username": "user0",  
      "DEGREE": "M50 60L148 60L148 20L674 20",  
      "INDEGREE": "M50 60L148 60L148 43L266 43L266 49L674 49",  
      "OUTDEGREE": "M50 60L148 60L148 20L674 20",  
      "RECIPROCITY": "M50 60L148 60L148 43L266 43L266 49L674 49",  
      "icon": "01.jpg"  
    }  
  ]  
/Codice degli altri utenti/  
}  
}
```

5.2.3. Database



Il modello dei dati utilizzato si basa su due entità principali: la sessione e il partecipante.

La sessione, indicata come meeting nel database, contiene tutte le informazioni di base della sessione come data e descrizione delle meeting. L'id è usato in oltre per recuperare tutti i dati legati alla sessione.

Il partecipante, indicato come people, contiene invece le informazioni e le anagrafiche sui partecipanti ai meeting.

È possibile associare i partecipanti a vari meeting, le informazioni di associazione sono contenute nella tabella ParticipantToMeeting, quale conserva alcune informazioni legate al complesso partecipante/sessione come ad esempio la posizione del partecipante nel workspace.

Gli eventi sono registrati nell'entità SpeakingEvents nella quale sono conservati un riferimento alla sessione, il timestamp dell'evento l'utente sorgente e (se esiste) l'utente ricevitore e il tipo di evento. Nel caso di eventi di tipo "SPEAKTOMANY" e "MOVE" sono utilizzate due tabelle ausiliarie, rispettivamente SpeakToManyDetails e movepeople, per l'inserimento di dati aggiuntivi.

Gli indici calcolati sono, invece conservate nell'entità UserStats (per gli indici legati all'utente), e GlobalStats (per gli indici globali).

Le note sono, invece, conservate nell'entità Notes.

Infine nel database sono conservati i dati relativi ai file delle icone dei soggetti, e del setting della stanza e collegati ai rispettivi Partecipanti e Sessioni che li utilizzano.

6. Test ergonomici

6.1 Ergonomia

Secondo l'associazione IEA (International Ergonomics Association) "ha come oggetto di studio l'attività umana in relazione alle condizioni ambientali, strumentali e organizzative in cui si svolge. Il fine è l'adattamento di tali condizioni alle esigenze dell'uomo, in rapporto alle sue caratteristiche e alle sue attività." Nata per studiare le norme di progettazione per la tutela della vita del lavoratore e per il miglioramento dell'efficienza e dell'affidabilità dei sistemi uomo-macchina, l'ergonomia ha allargato il proprio campo di applicazione alla progettazione di oggetti, servizi, ambienti di vita e di lavoro, che rispettino i limiti dell'uomo e di esso ne potenzino le capacità operative.

Uno dei campi di interesse dell'ergonomia è la Human Computer Interaction, ovvero lo studio di come avviene l'interazione tra l'uomo e l'interfaccia di un dispositivo (quale computer, telefonino ecc.), che ha come fine la progettazione di interfacce che siano usabili (l'usabilità è definita dall'ISO "l'efficacia, l'efficienza e la soddisfazione con le quali determinati utenti raggiungono determinati obiettivi in determinati contesti"), affidabili e siano di supporto all'attività umana.

6.2.1 Test di usabilità e Checklist

Il test di usabilità rappresenta uno dei metodi più efficaci di verifica dell'usabilità nei suoi diversi aspetti, compresa la misurazione. Consiste nell'esecuzione, da parte di un campione di utenti finali, di una serie di compiti o attività in interazione con il prodotto software da testare. Le difficoltà che gli utenti incontrano durante il test vengono analizzate per individuarne le cause e decidere le soluzioni per il miglioramento del prodotto.

Si attua in tre fasi fondamentali.

Nella fase di preparazione del test si definiscono gli obiettivi del test e le parti del prodotto da testare, si individuano e progettano i compiti che gli utenti dovranno svolgere durante il test e si selezionano gli utenti partecipanti.

La fase di esecuzione consiste nell'esecuzione dei compiti da parte degli utenti. Ciascun utente, singolarmente, esegue uno per volta i compiti progettati. La registrazione del test può essere supportata anche da strumenti semi-automatici di logging.

Nella fase di analisi vengono rivisti i dati del test, individuate le difficoltà dell'utente, le relative cause. Sulla base dei risultati dell'analisi si decidono le soluzioni di miglioramento da apportare al prodotto. Il test di usabilità è particolarmente indicato per la valutazione di prototipi funzionanti e di prodotti finali, prima del loro rilascio sul mercato. Il numero di utenti da coinvolgere in un test di usabilità varia in relazione alla quantità delle classi di utenza a cui è destinato il prodotto e ai risultati all'analisi costi/benefici. Può essere adottata la seguente regola: per ottenere dati qualitativi 5 utenti, se il prodotto è destinato ad una sola tipologia di utenti, 3-4 utenti per ogni tipologia, se il prodotto è destinato a due tipologie di utenti, 3 utenti per ogni tipologia, se le tipologie sono superiori a due;

per avere dati quantitativi, come ad esempio nel caso del calcolo delle metriche, sono necessari almeno a 20 utenti.

L'utilizzo di checklist per l'analisi ergonomica fa parte dei metodi detti di Usabilità, utilizzati per un'analisi qualitativa dell'usabilità di un prodotto.

Jakob Nielsen (*Usability Engineering*) e Ben Shneiderman (1987. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*) hanno diviso l'usabilità in 4 aree:

- Learnability: La facilità per un utente di comprendere il significato delle varie porzioni di interfaccia ad un primo utilizzo.
- Memorability: la facilità per un utente di ricordare il significato delle varie porzioni di interfaccia ai successivi utilizzi
- Errors: Quanti errori fa un utente, qual è la loro gravità e quanto facilmente, utilizzando l'interfaccia l'utente può porvi rimedio
- Satisfaction: Il grado di piacevolezza del design

Utilizzando questi 5 criteri è possibile creare una checklist, ovvero una serie di affermazioni alle quali gli utenti sono invitati a dare un giudizio di veridicità su una scala a 5 punti (scala Likert).

6.2.2 L'analisi dei movimenti oculari

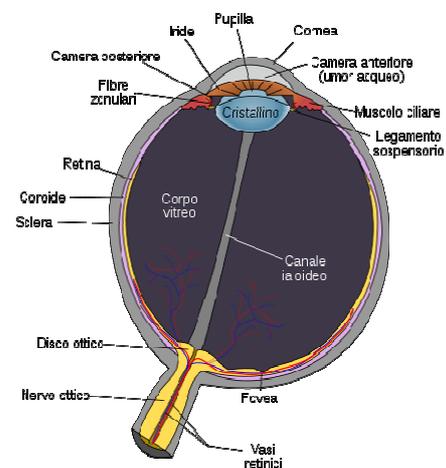
Tra le varie altre tecniche esistenti che si occupano di valutare l'usabilità di software e siti Internet l'analisi dei movimenti oculari è una delle più utilizzate e permette di ottenere moltissime informazioni sui processi cognitivi dell'utente.

Durante l'esplorazione di una schermata o la lettura di un testo (stampato o sul video) gli occhi compiono innumerevoli movimenti. E' infatti impossibile percepire distintamente tutto il contenuto dello schermo o della pagina che guardiamo con un solo "colpo d'occhio". Il colpo d'occhio ci dà semplicemente un'immagine generale del contenuto, ma per poterlo analizzare è necessario fissare con precisione ogni elemento che interessa (Weatherhead, James, 2005).

Questo perché solo la zona centrale dell'occhio (chiamata fovea) è abbastanza sensibile da percepire i dettagli (come possono essere le lettere di una parola). In effetti, solo ciò che guardiamo in modo diretto ci sembra chiaro, mentre ciò che percepiamo con "la coda dell'occhio" è molto più vago e meno preciso.

I movimenti degli occhi sono quindi destinati a piazzare in questa regione estremamente sensibile il contenuto di volta in volta ritenuto interessante.

Gli occhi possono compiere diversi tipi di movimenti, da molto lenti a molto rapidi. Nel caso dell'esplorazione del contenuto di una schermata di computer



(contenente le videate di un programma, delle pagine Web, ecc.) gli occhi presenteranno due comportamenti tipici fondamentali: delle saccadi e delle fissazioni. Le saccadi sono movimenti volti a posizionare il contenuto dello schermo nella zona di maggior risoluzione dell'occhio $400^\circ/\text{sec}$. Sono movimenti brevissimi, di durata tra i 20 e i 50 millisecondi ad altra velocità (circa $400^\circ/\text{secondo}$). Le fissazioni sono, invece, delle soste sui singoli elementi interessanti. Queste pause (la cui durata media è di circa un quarto di secondo ciascuna) sono volte al recupero e all'analisi del contenuto fissato, azioni che permettono di leggere un testo, interpretare un'immagine, ecc. I movimenti saccadici e le fissazioni si alternano, in modo che l'occhio si muova da un punto all'altro della pagina.

Registrare e analizzare il comportamento degli occhi di una persona che esplora le videate di un programma, oppure un sito web, mentre esegue un compito (ad esempio cercare una certa informazione, leggere il contenuto delle pagine, ecc.) può dare moltissime informazioni sui processi cognitivi: è possibile fare delle deduzioni sul livello di attenzione che la persona porta verso quello che sta osservando, sul suo modo di trattare le informazioni contenute nelle pagine, sulle sue strategie di esplorazione, e sui possibili problemi che può incontrare.

Un'analisi condotta in condizioni sperimentali non permette soltanto di sapere dove la persona ha guardato e dove no (ha osservato un certo pulsante? ha individuato una certa informazione?), ma anche di dedurre dati importanti sui suoi processi mentali (ha avuto difficoltà a trovare ciò che le interessava? Ha capito quello che stava leggendo?).

Le due fonti d'informazione più importanti per gli sperimentatori sono, appunto, le saccadi e le fissazioni dell'occhio. Le informazioni che si possono ricavare dall'analisi sono in grado di fornire delle informazioni circa la ricerca e il trattamento delle informazioni da parte dei soggetti i cui movimenti oculari sono stati registrati.

Attualmente sono disponibili tre tecniche che permettono di registrare i dati relativi ai movimenti degli occhi (Carpenter, Roger H.S.):

- *Elettroculografia*: per mezzo di elettrodi posti vicino agli occhi registra l'attività elettrica prodotta dalla loro rotazione. Le informazioni hanno una buona "risoluzione temporale" (si possono registrare molti dati al secondo) ma una scarsa "risoluzione spaziale" (è difficile dedurre con precisione la posizione dello sguardo). A causa di questi limiti, questa tecnica non è usata nello studio dell'ergonomia e dell'usabilità delle Interfacce.
- *Tecnica galvanometrica* o della "bobina sclerale": i movimenti oculari sono dedotti dalle variazioni nel campo magnetico in cui il soggetto è immerso. Sull'occhio del soggetto è posta una speciale lente a contatto contenente un filamento elettrico. La precisione dei dati è molto elevata ma questa tecnica è scarsamente utilizzata per lo studio dell'ergonomia e dell'usabilità delle Interfacce a causa della sua invasività (la lente a contatto è infatti particolarmente fastidiosa e irritante) e della necessità della presenza di un oftalmologo a ogni esperimento.
- *Tecnica del riflesso corneale*: inviando un piccolo fascio luminoso infrarosso al centro della pupilla, è possibile dedurre i movimenti dell'occhio dalle variazioni del riflesso rinviato, registrato da una telecamera posta sotto lo schermo del computer che la persona sta fissando oppure da un detettore montato in un casco che la persona indossa. Nel primo caso, nessun elemento dell'apparecchiatura tocca la persona (che quindi non è infastidita dalla sua

presenza), fatto che rende questa apparecchiatura particolarmente adatta agli esperimenti di ergonomia e usabilità delle Interfacce.

Per la sperimentazione eseguita in questo lavoro è stata utilizzata una macchina che utilizza la tecnica del riflesso corneale per la registrazione dei movimenti oculari. La macchina si presenta al soggetto come un normale monitor per PC, dentro il quale è però nascosta una telecamera ad infrarossi che permette di registrare, appunto, i movimenti che gli occhi compiono. Su tale display veniva perciò visualizzato il browser con il la web application da testare.

Per testare l'efficacia dell'interfaccia sono stati effettuati dei test eseguendo nel contempo la registrazione dei movimenti oculari assegnando dei compiti ai soggetti partecipanti. Di seguito è illustrato l'esperimento, il gruppo di soggetti utilizzati, la metodologia e i risultati ottenuti.

6.3 Scopo della sperimentazione

Gli scopi della sperimentazione erano duplici.

In primo luogo il test dell'interfaccia: la facilità di utilizzo delle varie schermate, la valutazione delle disposizioni dei compiti in maniera logica rispetto al reale utilizzo, la facilità di interpretazione di comandi, icone e statistiche, la stabilità del programma (in parte già precedentemente testata con una sessione di debugging), e la corretta disposizione degli elementi dell'interfaccia per una fruizione.

In secondo luogo dalla sperimentazione si voleva verificare l'effettiva utilità di uno strumento di questo tipo per l'analisi delle reti, testare la sua efficacia come strumento di raccolta dati programmatico e standardizzato, e verificare l'utilità degli indici considerati per l'analisi.

6.4 L'esperimento

L'esperimento è stato diviso in due parti.

Nella prima parte, dopo la visione di un tutorial di alcuni minuti, sono stati posti ai soggetti dei task da eseguire e sono stati registrati i movimenti oculari e il video delle operazioni che avvenivano a schermo tramite un eye-tracker.

I task eseguiti sono stati i seguenti:

1. Creare un nuovo meeting, inserendo il proprio nome nel campo "New Meeting Name", cliccando poi sul pulsante "Create"
2. Inserire il partecipante di nome "*nome partecipante*" nella sedia indicata:
 - "Marco" nella sedia in alto
 - "Luca" nella sedia di sinistra
 - "Giovanni" nella sedia di destra
 - "Stefano" nella sedia in basso
3. Avviare la registrazione degli eventi
4. Inserire i seguenti eventi comunicativi:
 - Marco parla con Luca
 - Luca parla con tutti

- Giovanni parla con Marco
 - Marco parla a Tutti
 - Giovanni parla con Marco
 - Giovanni parla con tutti
 - Stefano parla a Giovanni e Marco
 - Marco parla con Giovanni
 - Giovanni parla con tutti
 - Giovanni parla con tutti
5. Eseguire una serie di task utilizzando i pulsanti funzione (*UnDo,AddNote,ShowArcs*):
- Annullare l'ultimo evento inserito
 - Inserire una nuova nota marcando, così, l'istante di tempo
6. Identificare e fissare attraverso gli strumenti a disposizione (pulsante per identificare le connessioni della rete, grafici delle statistiche):
- Il leader della rete
 - La persona che ha parlato di più
 - La persona che ha ascoltato di più
 - La persona che ha interagito di più
7. Andare alla schermata di analisi.
8. Posizionare il marcatore del tempo intorno al minuto 1 in una delle due Timeline e inserisci una nuova nota. Modificare la descrizione e il commento della nota appena inserita

Nella seconda parte dell'esperimenti ai soggetti è stato somministrato un video contenente un'intervista ad un gruppo musicale ed è stato chiesto loro di registrare gli eventi comunicativi che avvenivano tra i partecipanti attraverso il software sviluppato. I soggetti sono stati divisi in due gruppi ai quali è stato chiesto di utilizzare il PC con mouse oppure il Tablet PC con pennino come dispositivo di puntamento (in questa seconda parte non sono registrati i movimenti oculari).

Al termine della visione del video sono state poste loro delle domande riguardanti i partecipanti al meeting, volte a testare l'efficacia e l'utilità degli indici utilizzati.

Al termine dell'esperimento è stato, infine, somministrata ai soggetti una checklist alle quali affermazioni i soggetti dovevano esprimere un giudizio (da 1, per niente d'accordo, a 5, totalmente d'accordo).

6.4.1 Le modifiche al software

Per una migliore verifica della disposizione dei comandi nell'interfaccia sono state testate due versioni del software, una prima descritta nel capitolo *Sviluppo* e denominata nel seguito *Versione A* ed una seconda, con alcune modifiche all'interfaccia chiamata in seguito *Versione B*.

Le modifiche introdotte nella *Versione B* rispetto alla *Versione A* sono:

- Barra di navigazione tra le schermate situata nella parte inferiore dello schermo anziché nella parte superiore

- Nella schermata di creazione del meeting scompare il pulsante trascina per spostare le icone dei partecipanti inserite: per effettuare questa azione occorre trascinare direttamente l'icona con



l'immagine del partecipante.

- Pulsanti per le azioni sui partecipanti più grandi
- Icone "UnDo", "AddNote" e "ShowArcs" nella parte bassa dell'area di lavoro anziché in alto a destra



- Pulsante "AddNote" nella schermata di analisi situato tra le Timeline anziché vicino alla lista delle note presenti nel sistema, al quale è applicata anche una diversa icona, invece dell'immagine la stessa della schermata precedente



6.4.2 Soggetti

Per l'analisi è stato utilizzato un campione di 13 soggetti di età media 24,54 (Dev. St. 5,09) Anni divisi in due gruppi: 7 soggetti assegnati al software *Versione A*, 6 soggetti assegnati al software *Versione B*. L'analisi ANOVA mostra che i due gruppi sono equivalenti rispetto all'età.

All'interno di ogni gruppo i soggetti sono stati divisi tra utenti esperti e utenti non esperti in base alla loro conoscenza di utilizzo del PC (giudicata da loro stessi su una scala da 1 a 4): in totale 6 utenti esperti e 7 utenti non esperti. L'analisi ANOVA mostra che i due gruppi sono equivalenti rispetto alla conoscenza del PC.

6.4.3 Risultati

6.4.3.1 *Analisi Checklist*

Verranno ora analizzati in dettaglio i risultati ottenuti dall'analisi dei giudizi riportati dai vari soggetti nella checklist finale, confrontando i giudizi dati dai soggetti che hanno utilizzato il software *Versione A* con quelli che hanno utilizzato la *Versione B*.

Le medie calcolate risultano sostanzialmente in linea con i risultati prefissi:

- *Apprendibilità e recupero delle informazioni*: il tutorial, secondo gli utenti, risulta utile per l'apprendimento del software e i comandi risultano facili da ricordare.
-

Apprendibilità e recupero delle informazioni	Media A	STD	Media B	STD
1. Le istruzioni del tutorial sono state chiare	2,78	1.39	4,25	0.5
2. Seguendo il tutorial ho appreso abbastanza per cominciare ad utilizzare il software	3,00	1.11	4,25	0.5
3. Ricordo come creare una sessione di lavoro	4,33	1.11	5,00	0
4. Ricordo come inserire gli utenti nell'area di lavoro	4,67	0.7	5,00	0
5. Ricordo come assegnare il nome agli utenti	4,78	0.67	5,00	0
6. Ricordo come avviare la registrazione del meeting	4,00	1.32	4,75	0.44
7. Ricordo come disegnare la comunicazione tra due partecipanti	4,56	1.01	5,00	0
8. Ricordo come disegnare una comunicazione da una a più persone	4,56	0.88	4,50	1
9. Ricordo come disegnare una comunicazione da una persona a tutti	4,56	1.33	5,00	0

- *Navigabilità e Design*: la divisione logica tra le schermate è giudicata ben progettata, la scelta dei colori e la disposizione dei comandi risulta buona; la veste grafica è giudicata più professionale nella prima versione. La navigazione nella timeline e l'aggiunta di note nella schermata "ANALYSIS" presenta però alcune problematiche, nella prima versione del software ottiene infatti un punteggio di 2.33/5, risolte probabilmente nella seconda versione con lo spostamento di alcuni comandi dell'interfaccia (Centramento della barra per la selezione degli indici visualizzati e spostamento del pulsante "Aggiungi Nuova Nota")

Navigabilità e design	Media A	STD	Media B	STD
10. Ho trovato le combinazioni di colori piacevoli	3,22	1.09	4,00	0.81
11. La grafica mi è stata spesso d'aiuto	3,33	1.22	4,00	0
12. La veste grafica è professionale	3,67	1.12	2,75	0.5
13. Le componenti della schermata iniziale sono disposte in modo ordinato	4,11	0.78	4,25	0.5
14. Le componenti della schermata di lavoro sono disposte in modo ordinato	4,11	0.6	4,50	0.2
15. Le componenti della schermata di resoconto sono disposte in modo ordinato	4,11	0.92	3,75	0.95
16. Le dimensioni dei comandi di disegno sono adeguate	3,56	1.23	4,25	0.41
17. Le dimensioni degli altri comando d'interfaccia sono	3,67	0.72	3,75	0.5

adeguate				
18. è facile spostare le icone dei partecipanti nell'area di lavoro	4,67	0.71	4,75	0.5
19. è facile aggiungere nuovi partecipanti	4,67	0.5	4,50	1
20. è facile aggiungere le note nella schermata di analisi	2,78	1.39	3,00	0.81
21. è facile modificare il contenuto delle note nella schermata di analisi	3,67	1.41	3,50	0.58
22. è facile navigare nella timeline della pagina di analisi	2,33	0.71	4,25	0.5

- *Visibilità e comprensibilità*: Le informazioni essenziali allo svolgimento dei task sono state giudicate facilmente visibili e comprensibili in entrambe le versioni

Visibilità e comprensibilità	Media A	STD	Media B	STD
23. sapevo cosa cliccare per eseguire una determinata azione	3,44	0.88	4,00	0
24. i nomi dei comandi erano adeguati	3,44	1.13	3,75	0.95
25. è stato facile trovare i comandi per passare da una schermata all'altra	3,67	1.32	4,00	0.81
26. ho ricevuto un adeguato feedback per i comandi di disegno	3,44	0.53	4,00	0.82
27. gli effetti delle mie azioni nel software erano visibili	4,22	0.97	4,25	0.96
28. è stato facile identificare di volta in volta i partecipanti al meeting coinvolti	4,00	1.17	4,50	0.58

- *Gestione degli errori*: l'interfaccia non ha portato a commettere errori e ha permesso una facile correzione degli errori commessi

gestione degli errori	Media A	STD	Media B	STD
29. l'interfaccia mi ha portato a commettere errori	2,33	1	2,50	1
30. l'interfaccia mi permette di rimediare agli errori commessi	3,44	1.13	4,00	0.81
31. La posizione dei comandi di disegno induce a commettere errori	2,00	1.18	2,25	1.25
32. Gli errori commessi hanno compromesso il risultato finale dei compiti	2,89	1.36	2,25	0.5

- *Efficacia del mapping*: la situazione reale e le azioni compiute in un meeting sono mappate efficacemente all'interno del software

Efficacia del mapping	Media A	STD	Media B	STD
33. Le conversazioni disegnate sono fedeli ai reali turni di conversazione	3,78	0.79	4,00	0
34. Il resoconto finale dell'applicazione rispecchia la realtà del meeting	3,89	0.93	4,00	0

35. L'interfaccia è efficace nel rappresentare la realtà del meeting	4,11	0,93	4,00	0
--	------	------	------	---

(I test ANOVA sulle affermazioni si possono trovare in appendice).

6.4.3.2 Analisi del questionario finale sull'efficacia degli algoritmi

Viene di seguito riportato il questionario somministrato ai soggetti dopo la seconda parte dell'esperimento, riguardante la visione del video e la registrazione degli eventi comunicativi.

Basandoti esclusivamente sugli elementi messi a disposizione dal programma (pulsante per identificare le connessioni della rete, statistiche in tempo reale...) individua:

- I leader della rete _____
 - i. Che statistica/funzione ti ha aiutato maggiormente per l'identificazione di questa informazione? _____
 - ii. La ritieni coerente con l'impressione che hai avuto durante la registrazione degli eventi (da 1 a 5)? _____

- La persona che hanno parlato di meno _____
 - i. Che statistica/funzione ti ha aiutato maggiormente per l'identificazione di questa informazione? _____
 - ii. La ritieni coerente con l'impressione che hai avuto durante la registrazione degli eventi (da 1 a 5)? _____

- Le due persone che hanno interagito di più tra loro _____
 - i. Che statistica/funzione ti ha aiutato maggiormente per l'identificazione di questa informazione? _____
 - ii. La ritieni coerente con l'impressione che hai avuto durante la registrazione degli eventi (da 1 a 5)? _____

- Il numero di messaggi scambiati _____

- Le statistiche visualizzate hanno un significato comprensibile? (Da 1 a 5) _____

- Le consideri utili (da 1 a 5)? _____

6.4.3.3 Efficacia degli indici

Le prime tre domande poste chiedevano di identificare, i nodi della rete che avessero le caratteristiche indicate (il leader della rete, il nodo che ha comunicato meno, e i 2 nodi che hanno interagito di meno col gruppo) utilizzando solo le informazioni messe a disposizione dal software, e di confrontarle poi con l'opinione personale che l'utente aveva delle relazioni tra le persone del video. I risultati, riportati nei grafici, hanno mostrato una generale uniformità nelle risposte tra i vari soggetti pure con una differenza di numero di eventi inseriti, sintomo che la raccolta dei dati e il conseguente calcolo degli indici avviene in maniera sistematica, nonostante la differenza di conoscenza nell'uso del computer e dei concetti relativi all'analisi delle reti sociali dei soggetti. Inoltre, a giudizio degli utenti le informazioni calcolate dal software risultano coerenti con le opinioni personali sulle relazioni della rete.

Infine, i dati raccolti dagli utenti risultano coerenti anche con i risultati attesi nella rete, calcolati a priori.

6.4.3.4 Comprensibilità delle statistiche

È stato inoltre richiesto di indicare agli utenti quale caratteristica del software è stata più utile per l'individuazione dei nodi richiesti, anche in questo caso, le risposte si sono rivelate coerenti con i risultati attesi, sono state cioè utilizzati gli indici corretti per l'identificazione delle caratteristiche.

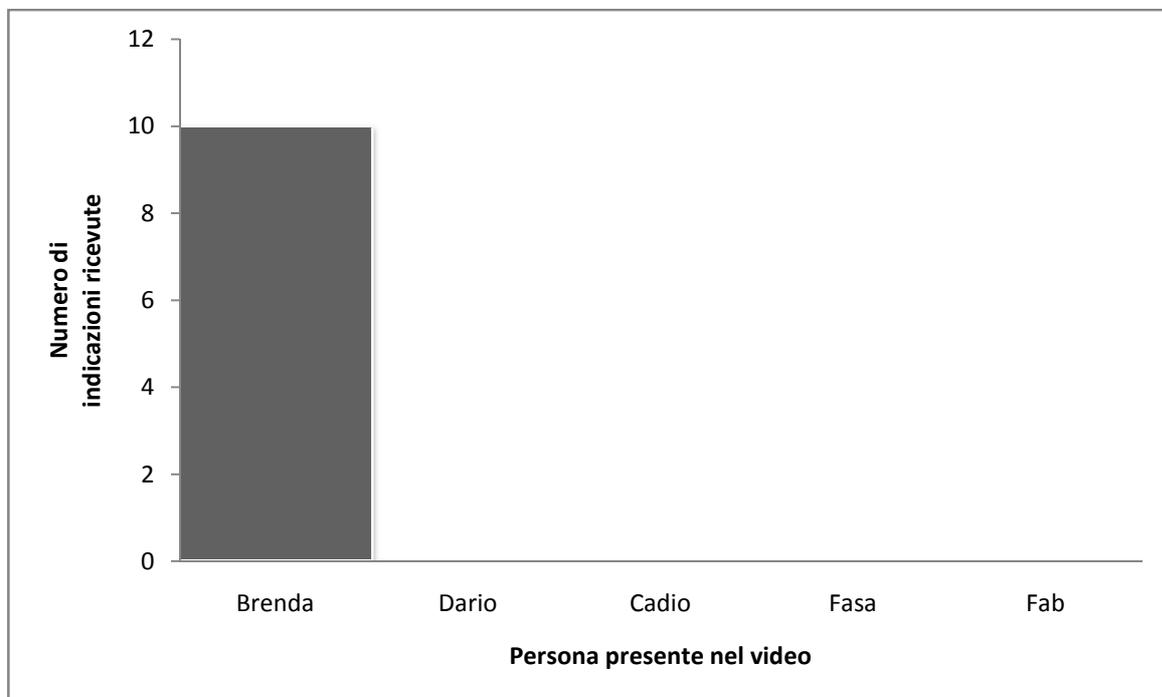


Grafico 1. "Chi è il Leader della rete?"

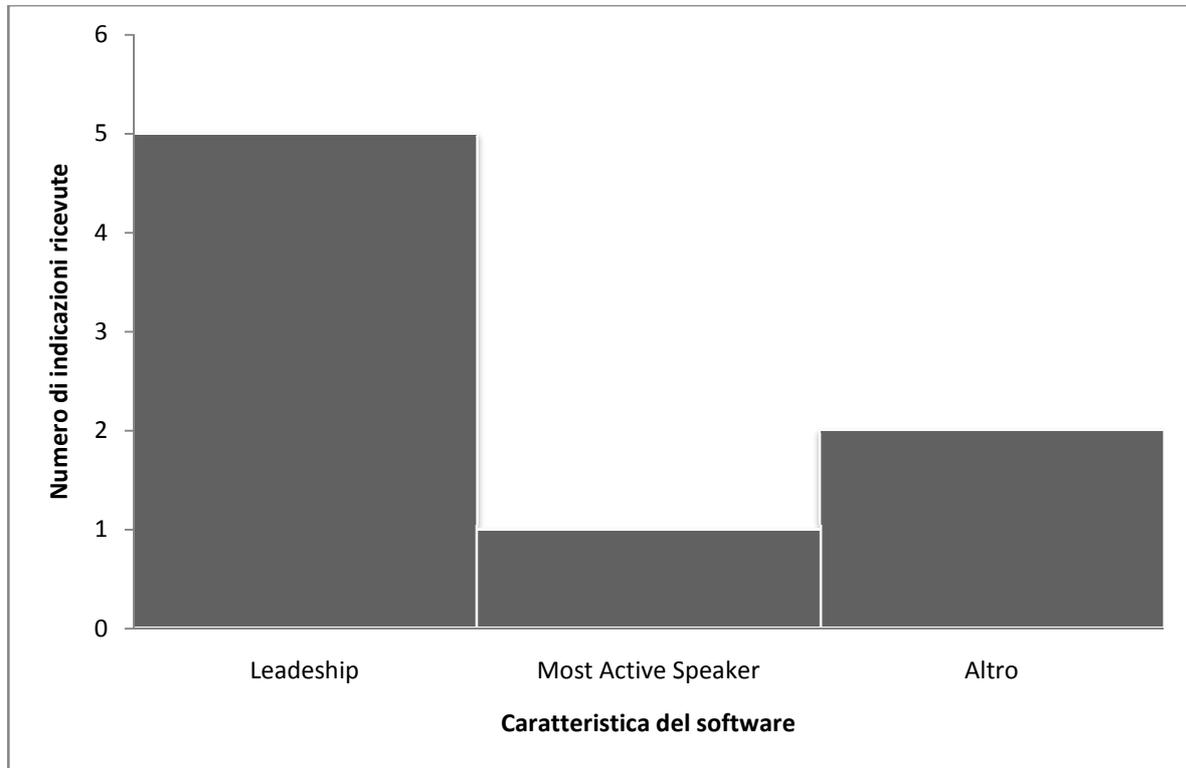


Grafico 2 "Che statistica/funzione ti ha aiutato maggiormente per l'identificazione del leader della rete?"

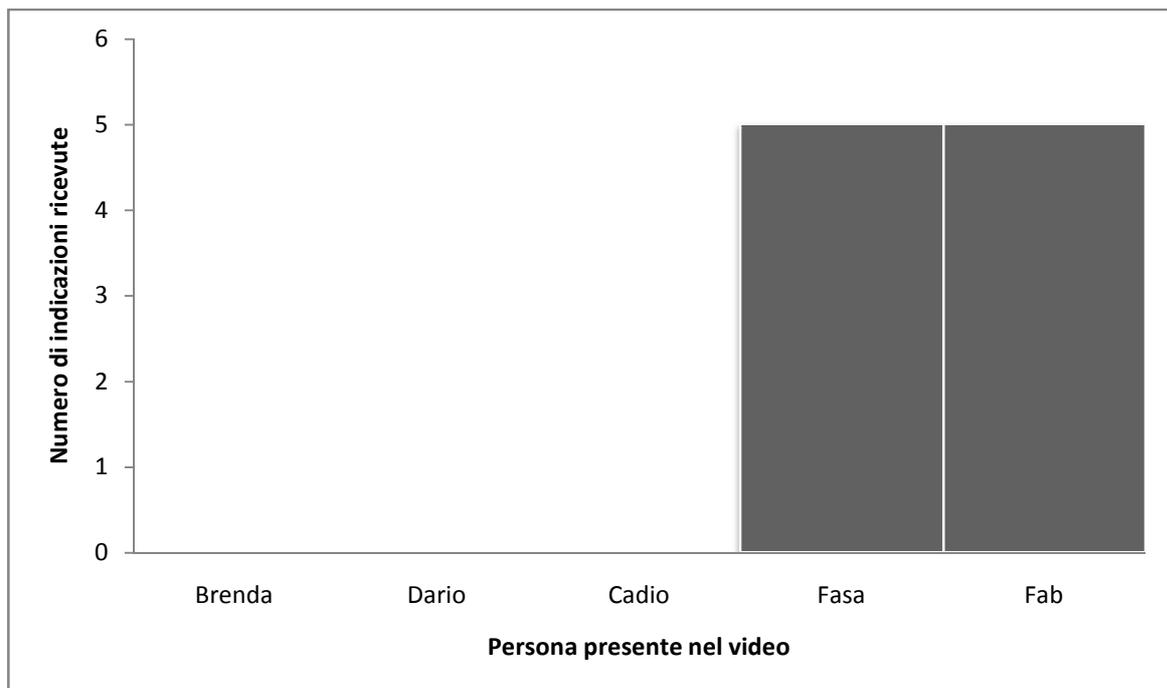


Grafico 3 Chi è la persona che ha parlato di meno?

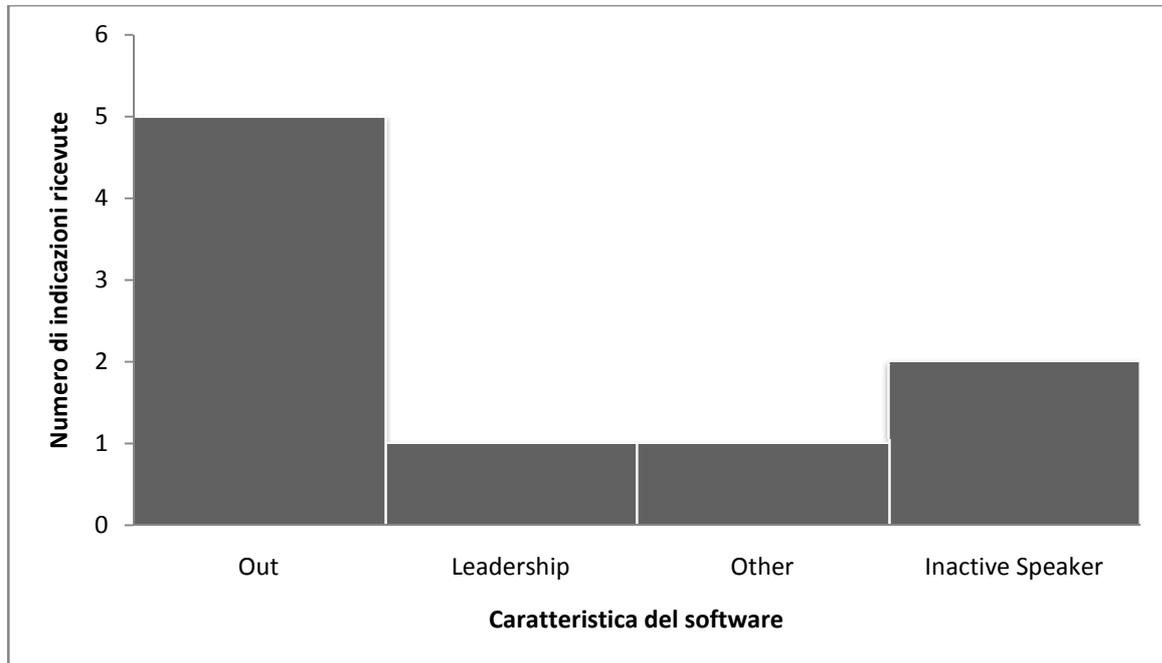


Grafico 4 "Che statistica/funzione ti ha aiutato maggiormente per l'identificazione della persona che ha parlato di meno?"

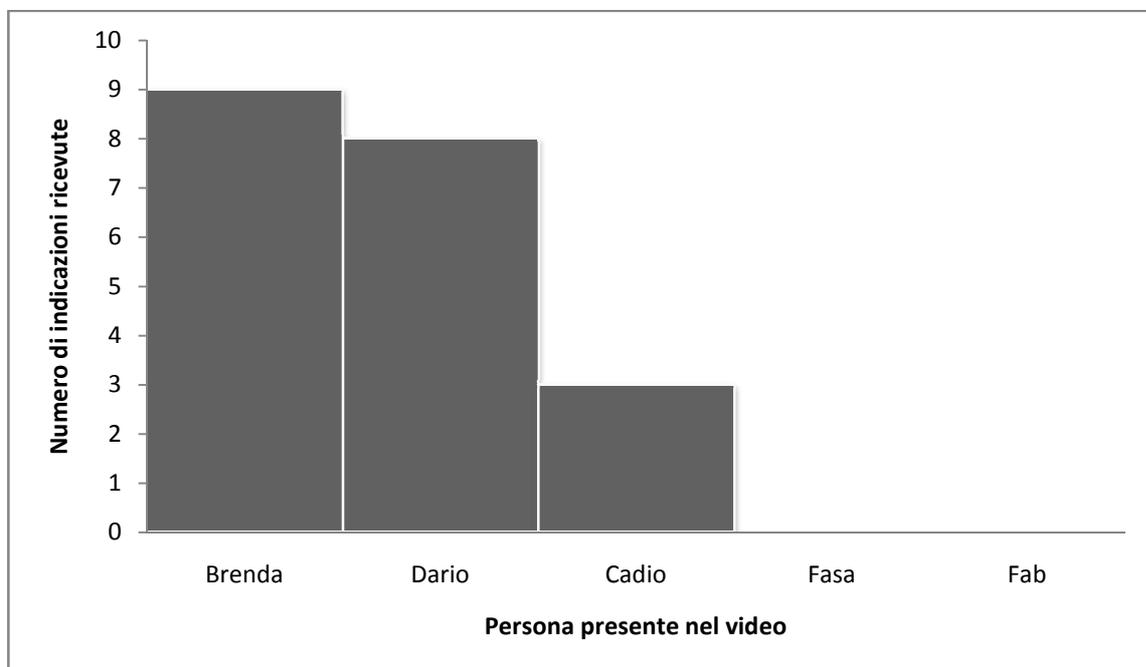


Grafico 5 "Quali sono le due persone che hanno interagito di più tra loro?"

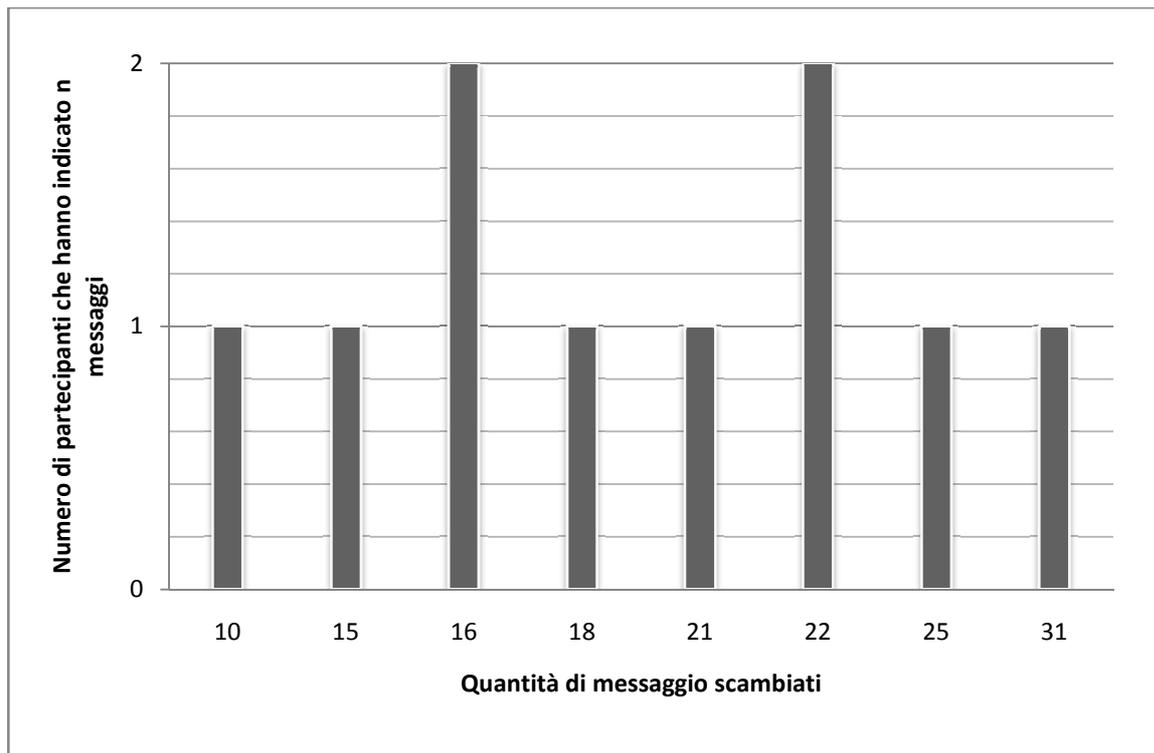


Grafico 1 Il numero di messaggi registrati dagli utenti

6.4.3.5 Comprensibilità e utilità

A giudizio degli utenti le statistiche sono risultate comprensibili e utili; è interessante notare come, nonostante vengano presentate nella stesso modo sia su piattaforma TabletPen che su piattaforma con Mouse e non presentino elementi interattivi, l'utilità ha ricevuto un punteggio molto superiore dagli utenti che hanno utilizzato il PC con TabletPen rispetto agli altri. Questo secondo gli studi di Fogg (Credibility and Computing Technology, 1999) è da imputare al divertimento maggiore degli utenti nell'utilizzo di una tecnologia (Il Tablet PC) non convenzionale.

	Tablet PC		PC con Mouse	
	Media	SDV	Media	SDV
Coerenza secondo gli utenti delle informazioni ricavate sul Leader	4,33	0.89	4,25	0.58
Coerenza secondo gli utenti delle informazioni ricavate sulla persona che ha parlato meno	4,17	0.79	4,75	1
Coerenza secondo gli utenti delle informazioni ricavate sull'interazione tra i partecipanti	4,17	0.75	4,75	0.28
Comprensibilità secondo gli utenti delle informazioni	3,83	0.97	3,50	0
Utilità secondo gli utenti delle informazioni	4,33	0.89	3,00	0.57

(In Appendice sono riportati i test ANOVA sul questionario)

6.4.4 Analisi dei task

6.4.4.1 Task effettuati/ non effettuati

Non tutti i task dati sono stati eseguiti correttamente dagli utenti. In tabella sono riportati le percentuali di task eseguiti correttamente dai singoli utenti. Si può notare che il numero di task non effettuati o non riusciti è inferiore nella seconda versione del software, sintomo che l'interfaccia risulta migliore. Con un'analisi più in profondità rispetto al contenuto dei task si può nota che la maggior dei task non effettuati risulta essere quella legata all'aggiunta delle note; le modifiche introdotte nella seconda versione hanno notevolmente migliorato l'interfaccia. Suddividendo inoltre gli utenti in esperti e non esperti (utilizzando il giudizio sul loro gradi preparazione dato nel questionario) si nota un minor numero di task sbagliati negli utenti non esperti.

Tipologia Task	Versione A		Versione B	
	<i>Esperti</i>	<i>Non esperti</i>	<i>Esperti</i>	<i>Non esperti</i>
Creazione Meeting	100%	100%	100%	100%
Inserimento Partecipanti	100%	100%	100%	100%
Avvio Registrazione	100%	80%	100%	67%
Registrazione eventi	88%	80%	100%	97%
Uso dei pulsanti dell'interfaccia (Undo, NewNote, ShowArcs, NextPage)	60%	67%	50%	33%
Analisi Dati Offline	44%	40%	100%	75%

6.4.4.2 Tempi di esecuzione

Nel grafico sottostante si può notare che è presente una correlazione bassa tra numero di task eseguiti e tempo di esecuzione (coefficiente di correlazione di Pearson = 0.29); il numero di task eseguiti correttamente non dipende perciò dalla velocità con cui il task è stato compiuto.

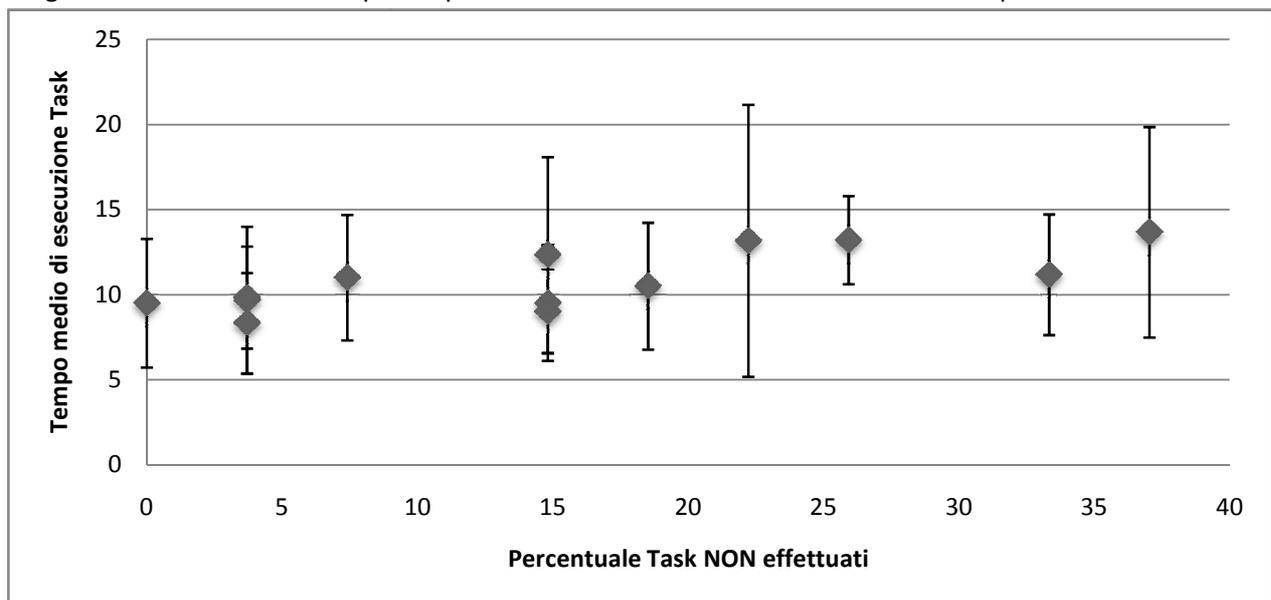


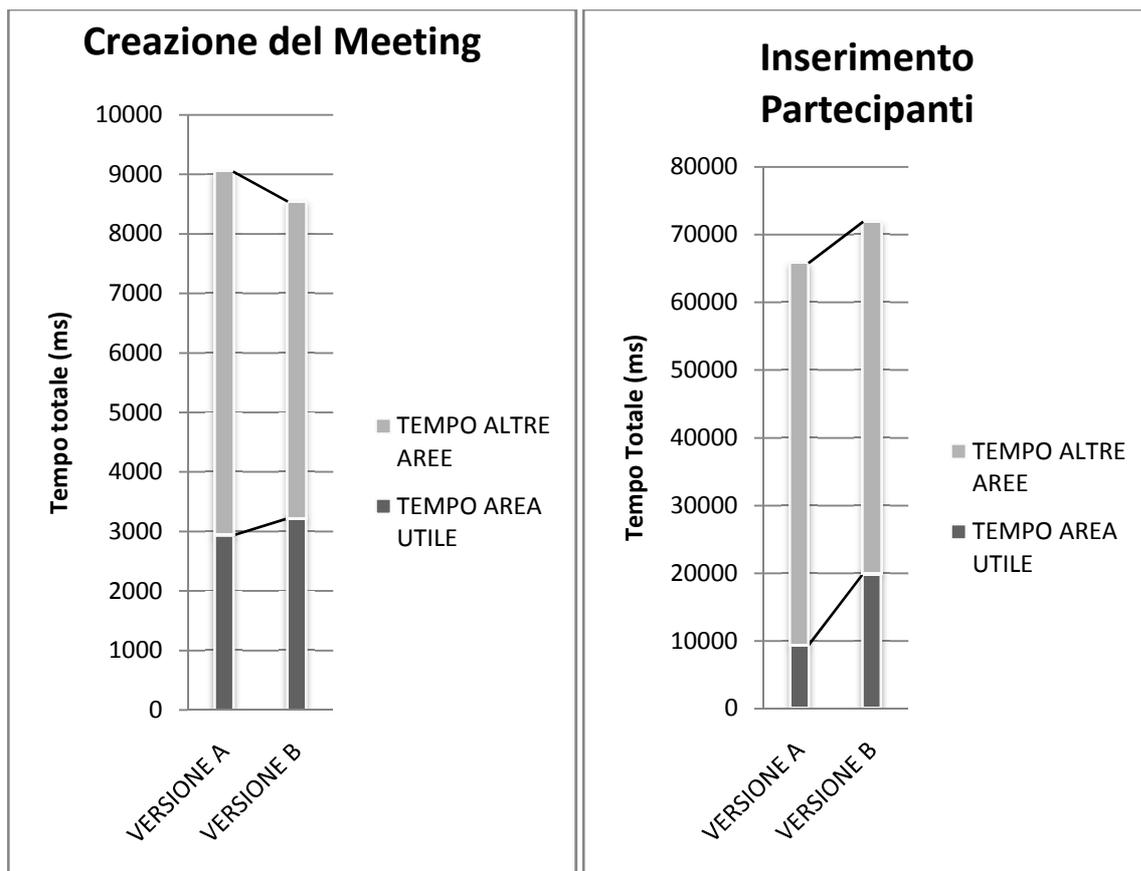
Grafico 2 Correlazione numero di task non effettuati / media tempi esecuzione

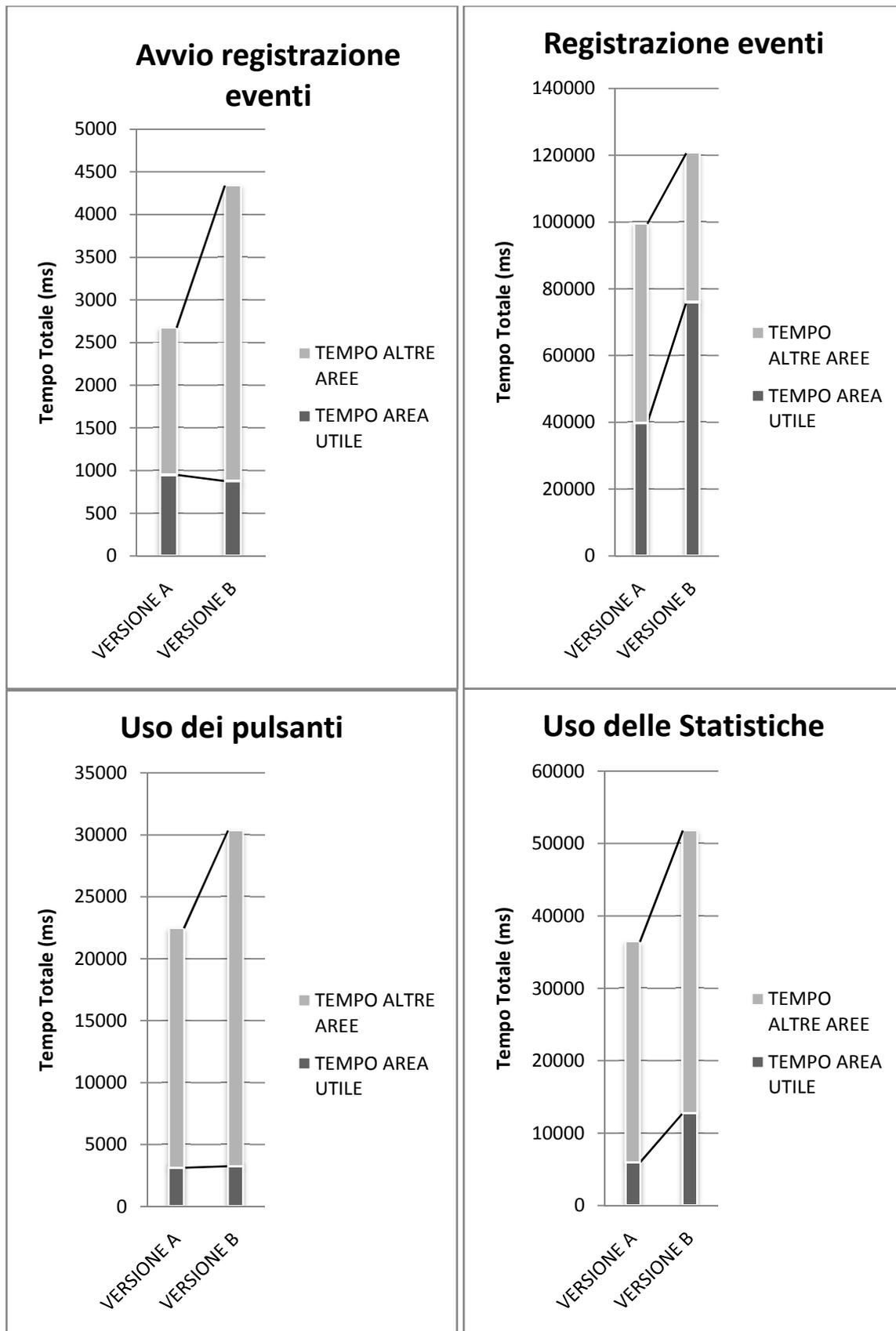
6.4.5 Analisi dei Blink

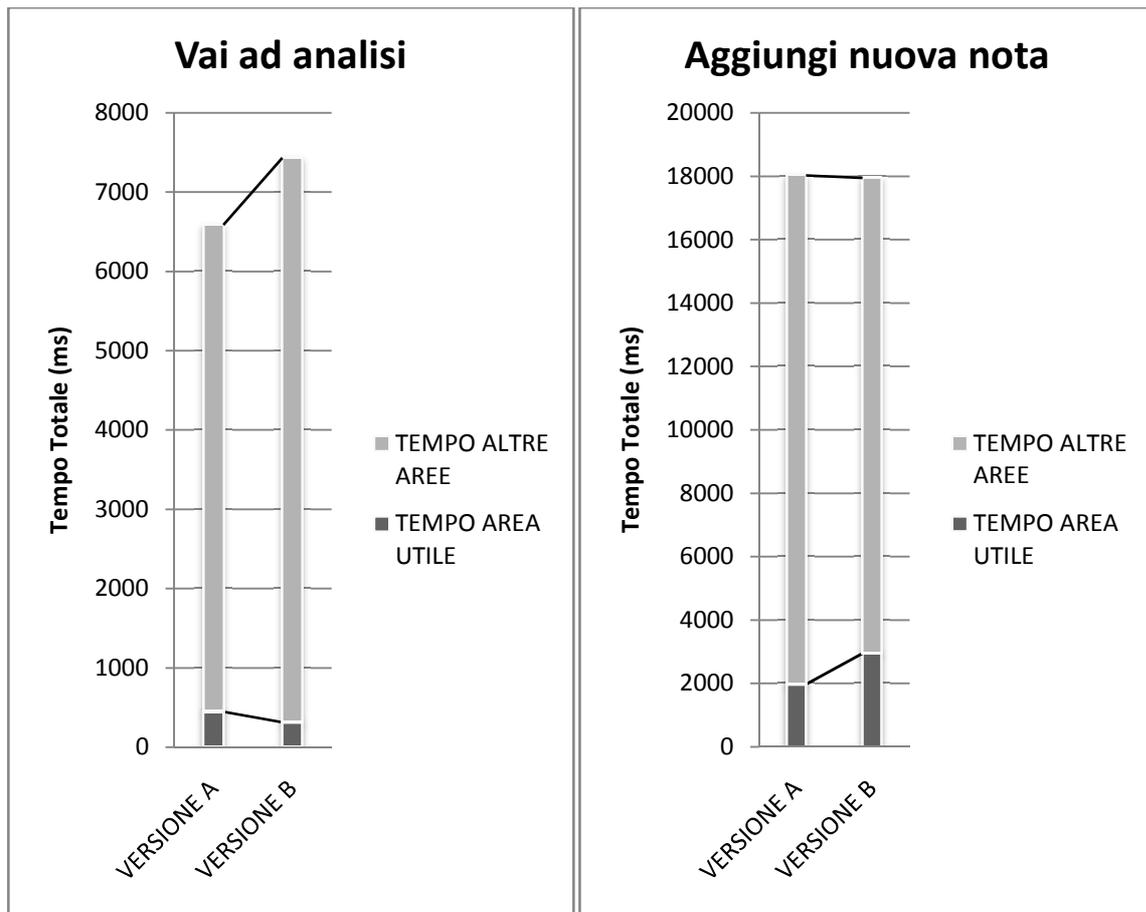
L'analisi dei blink non mostra differenze significative in nessuna delle categorie analizzate. Il sistema di eyetracking utilizzato si è rivelato, tuttavia, inadatto alla rilevazione dei blink delle palpebre a causa del basso framerate che utilizza per la cattura delle informazioni .

6.4.6 Analisi dei movimenti oculari

Nei grafici delle pagine successive sono riportati i risultati ottenuti tramite la tecnica dell'analisi dei movimenti oculari, suddivisi per task. In particolare per ogni gruppo di task considerato sono presenti i grafici contenenti i tempi totali di fissazione, il numero totale di fissazioni e il tempo medio di fissazione per ogni singolo task, suddivisi per le aree di interesse per l'analisi. Sono riportati, inoltre i grafici del tempo totale di esecuzione dei task suddivisi tra tempo speso in aree utili e tempo speso in aree inutili alla buona riuscita del task.







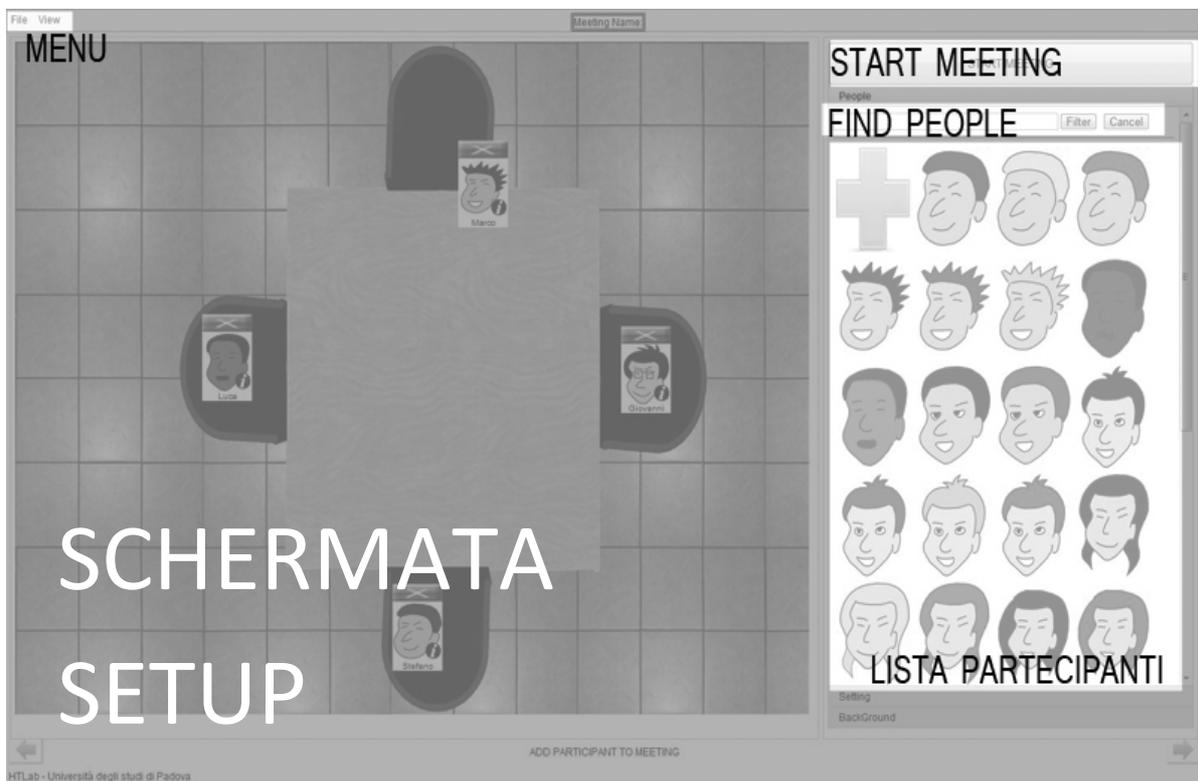
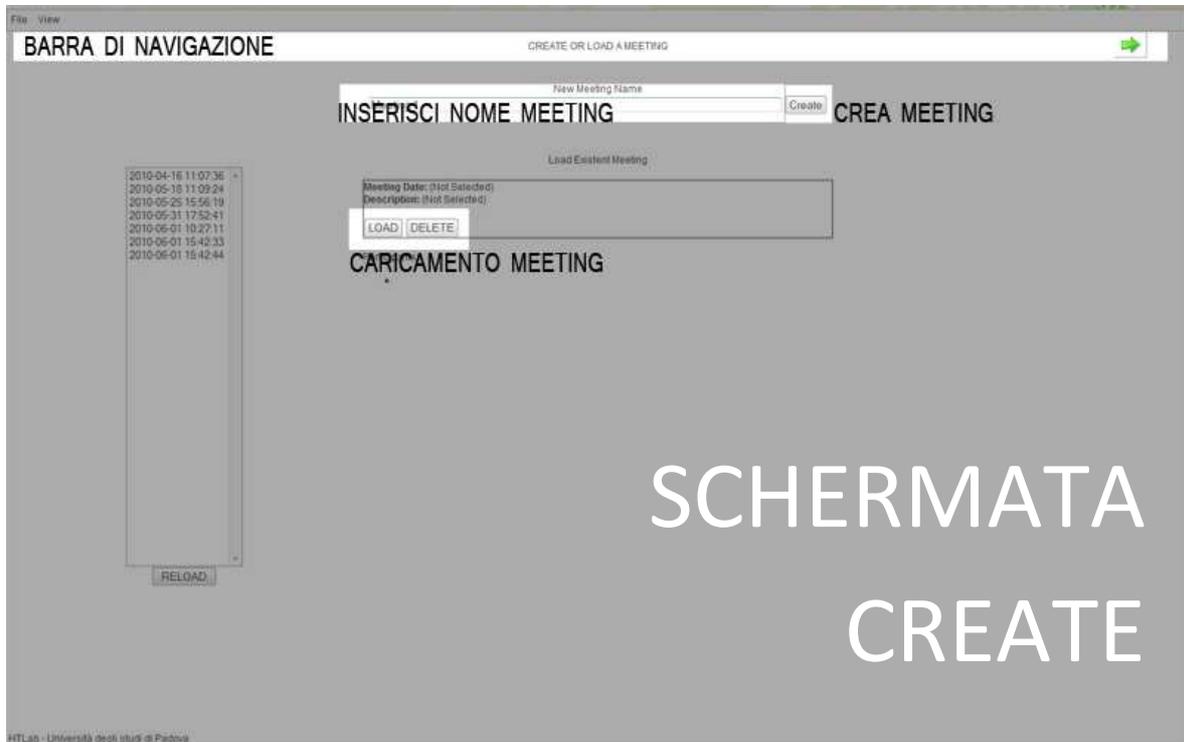
Nel caso dei task “Avvio registrazione eventi”, “Uso dei pulsanti”, “Uso delle statistiche” e “Vai ad analisi”, a fronte di un aumento del tempo totale di esecuzione del task nella Versione B rispetto alla Versione A si può notare, come il tempo speso all’interno delle aree utili all’esecuzione del task sia risultato simile in entrambe le versioni. Le risorse cognitive vengono quindi, nella versione B, disperse maggiormente in aree inutili al task.

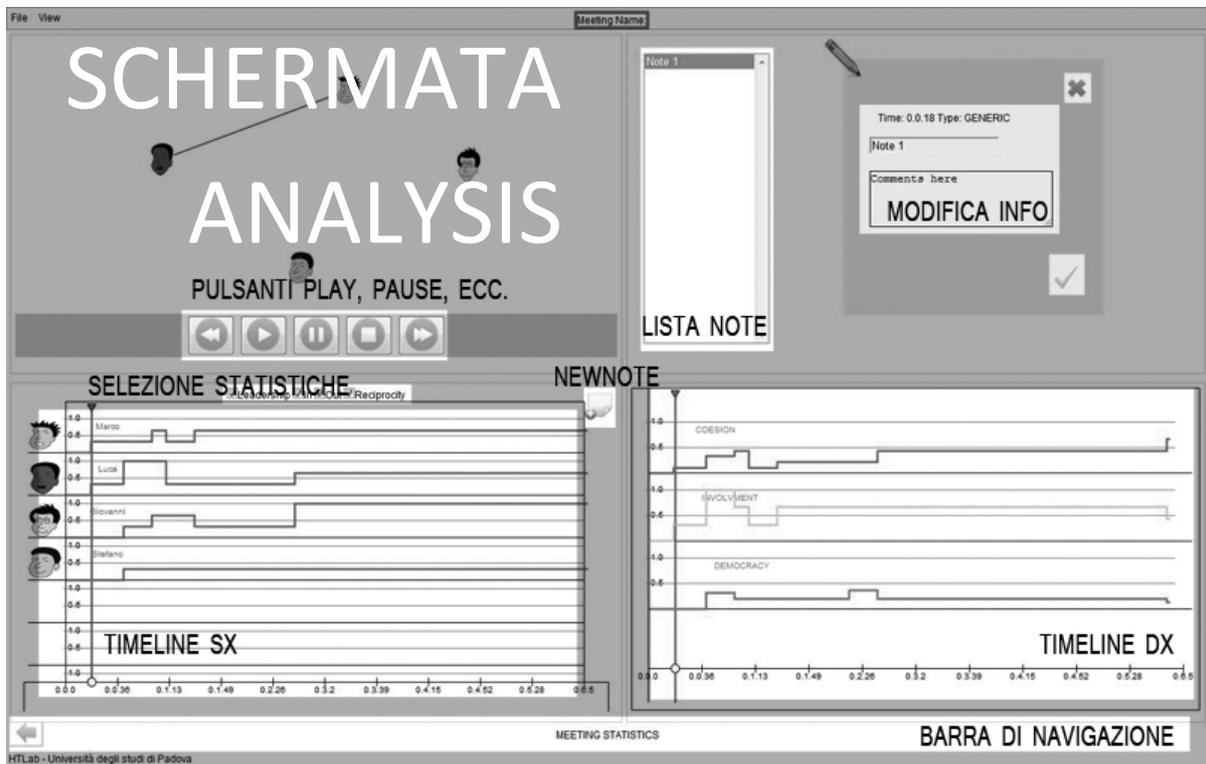
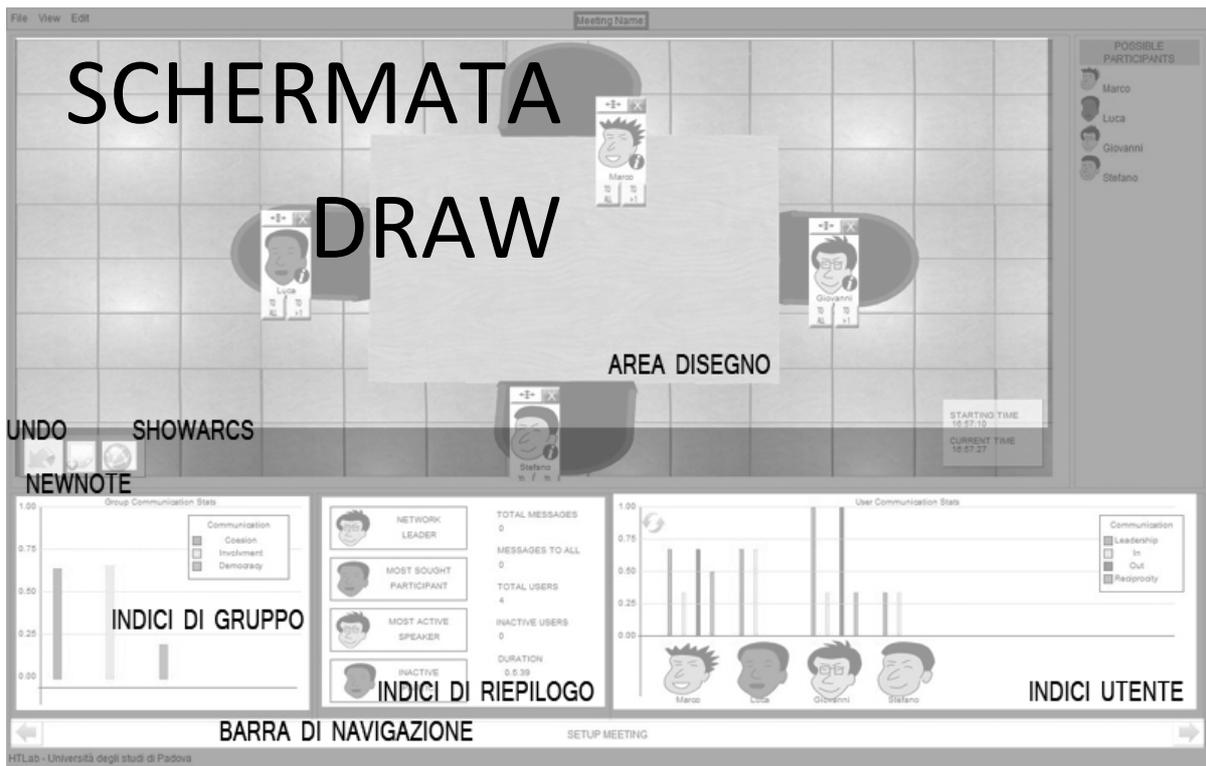
Nel caso dei task “Inserimento Partecipanti” e “Registrazione eventi”, il test ANOVA evidenzia, invece, una sostanziale equivalenza del tempo totale di esecuzione contro un aumento significativo del tempo speso in aree utili allo svolgimento del compito. La nuova disposizione grafica è quindi cognitivamente più funzionale, e perciò da preferirsi.

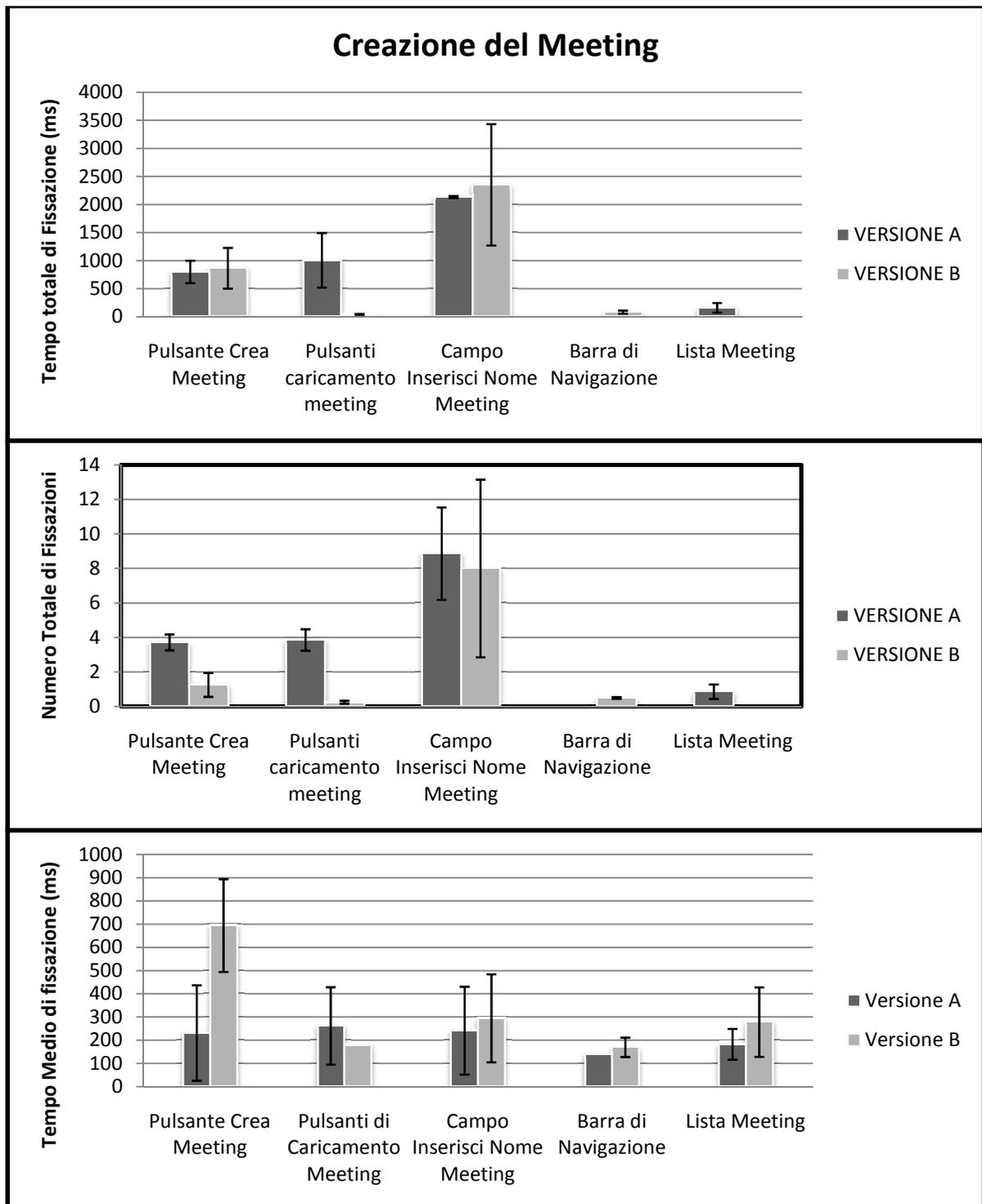
Nei rimanenti casi le due Versioni del software, per quanto riguarda i tempi di esecuzione dei task, sono equivalenti

Infine, i task “Uso dei pulsanti”, “Uso delle statistiche” e “Vai ad analisi”, mostrano una differenza elevata tra il tempo impiegato in aree utili ed il tempo impiegato in altre aree. I comandi per l’esecuzione dei task indicati sono, perciò, difficili da individuare.

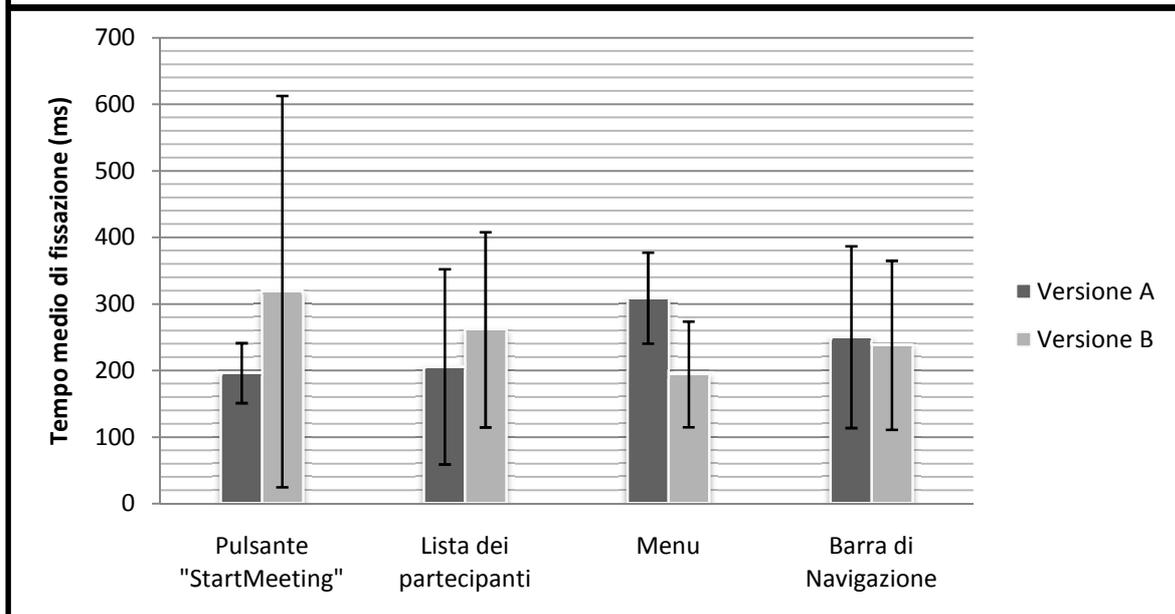
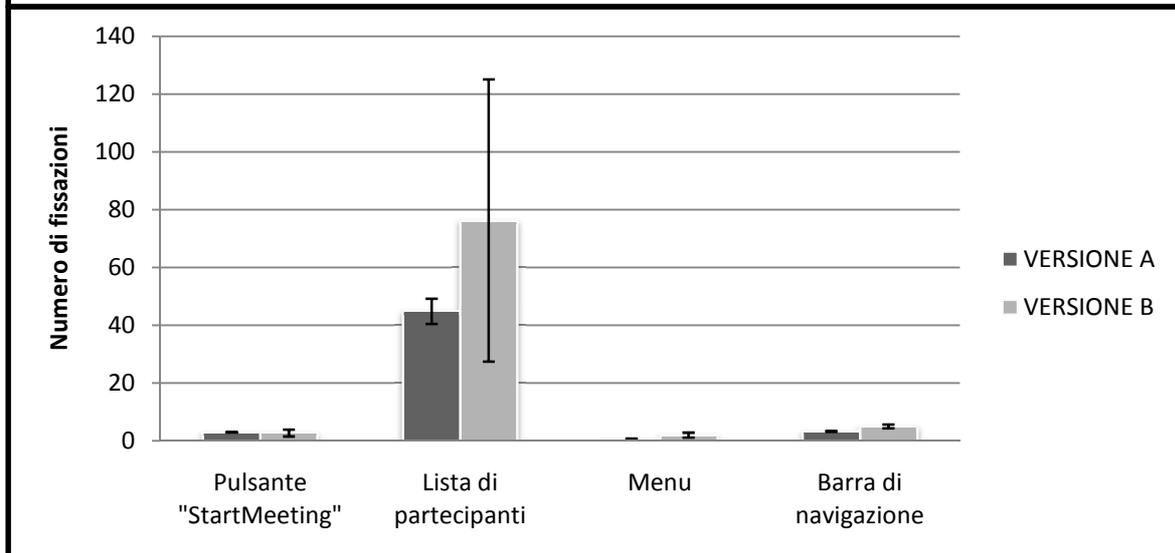
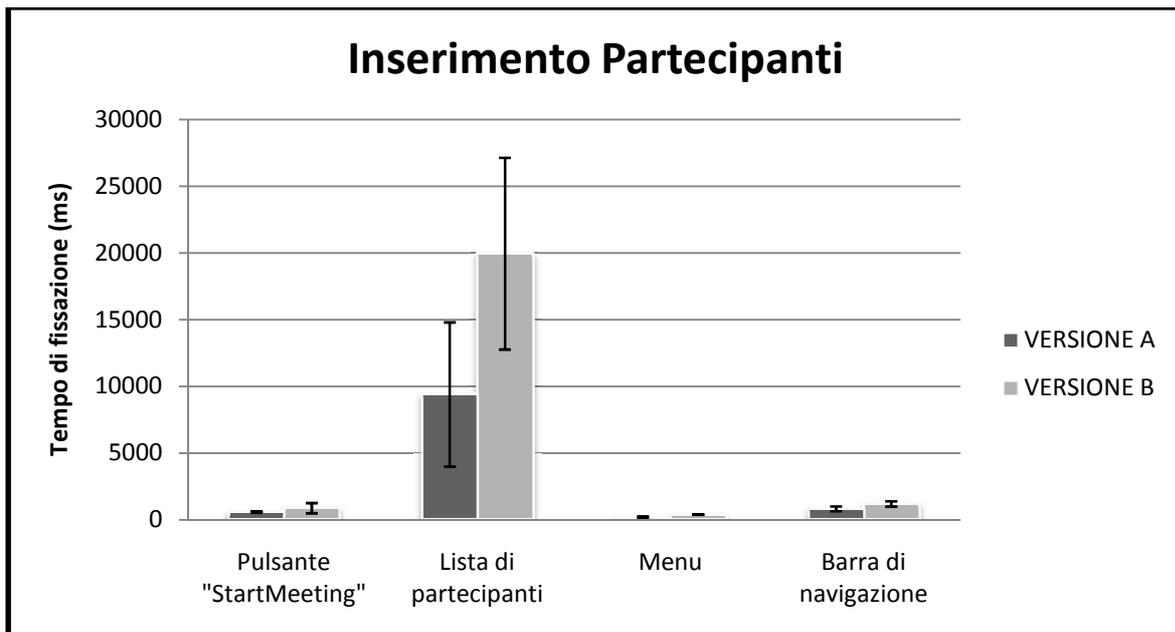
Suddivisione Aree di Interesse

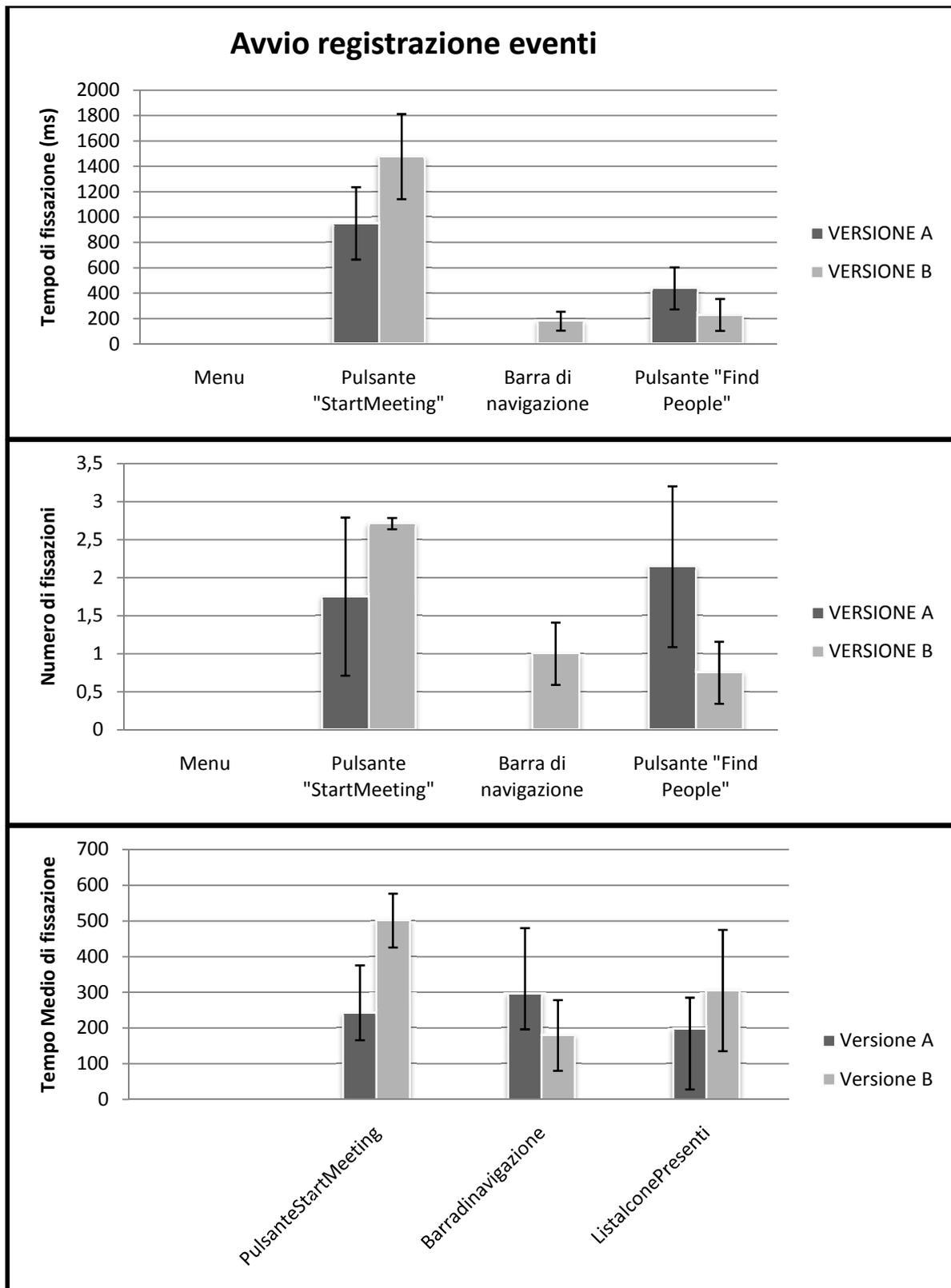




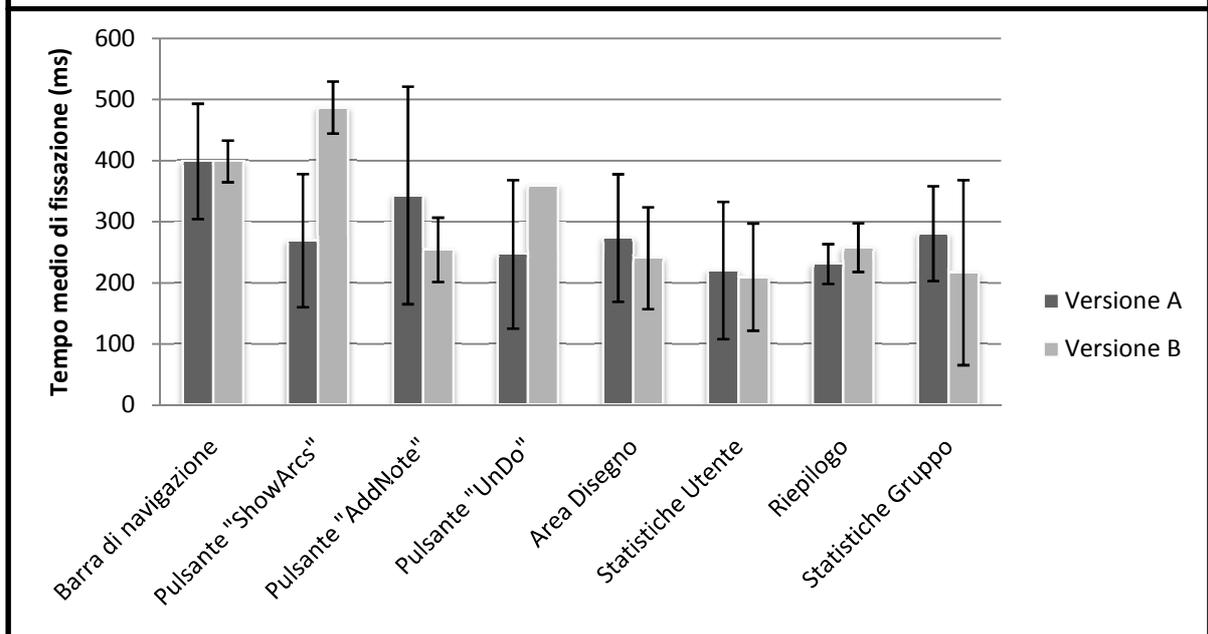
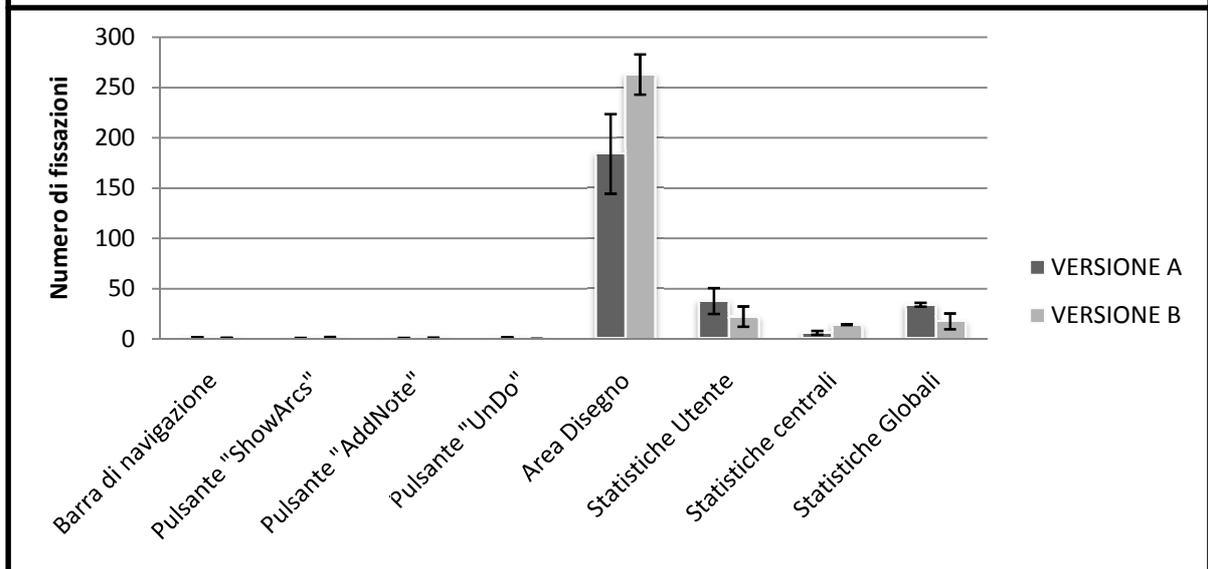
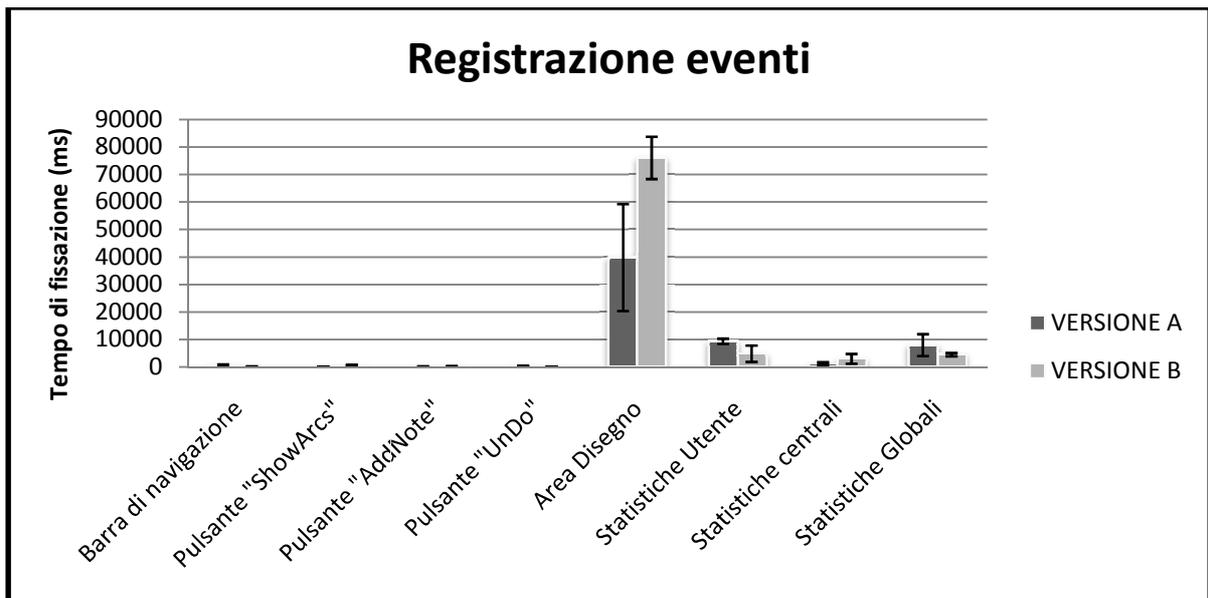


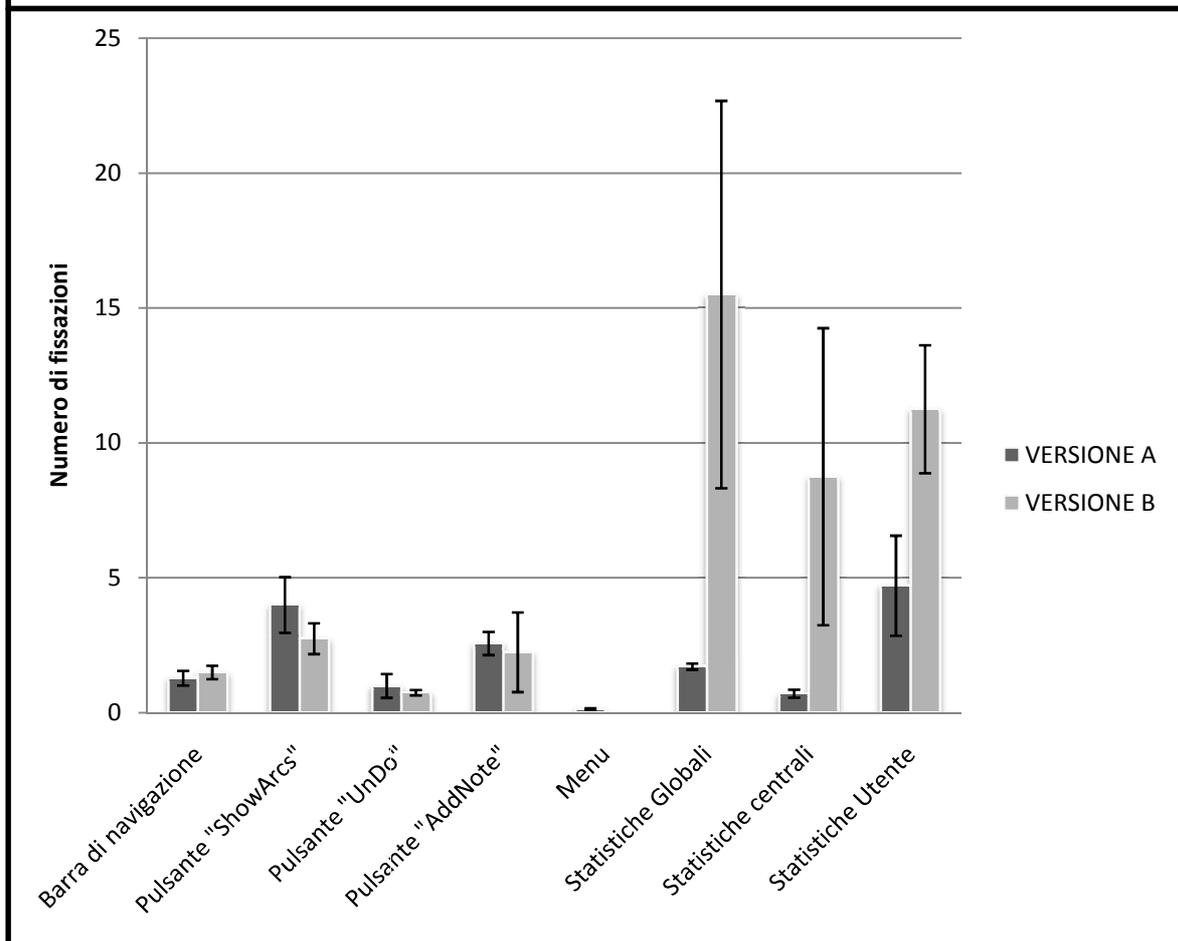
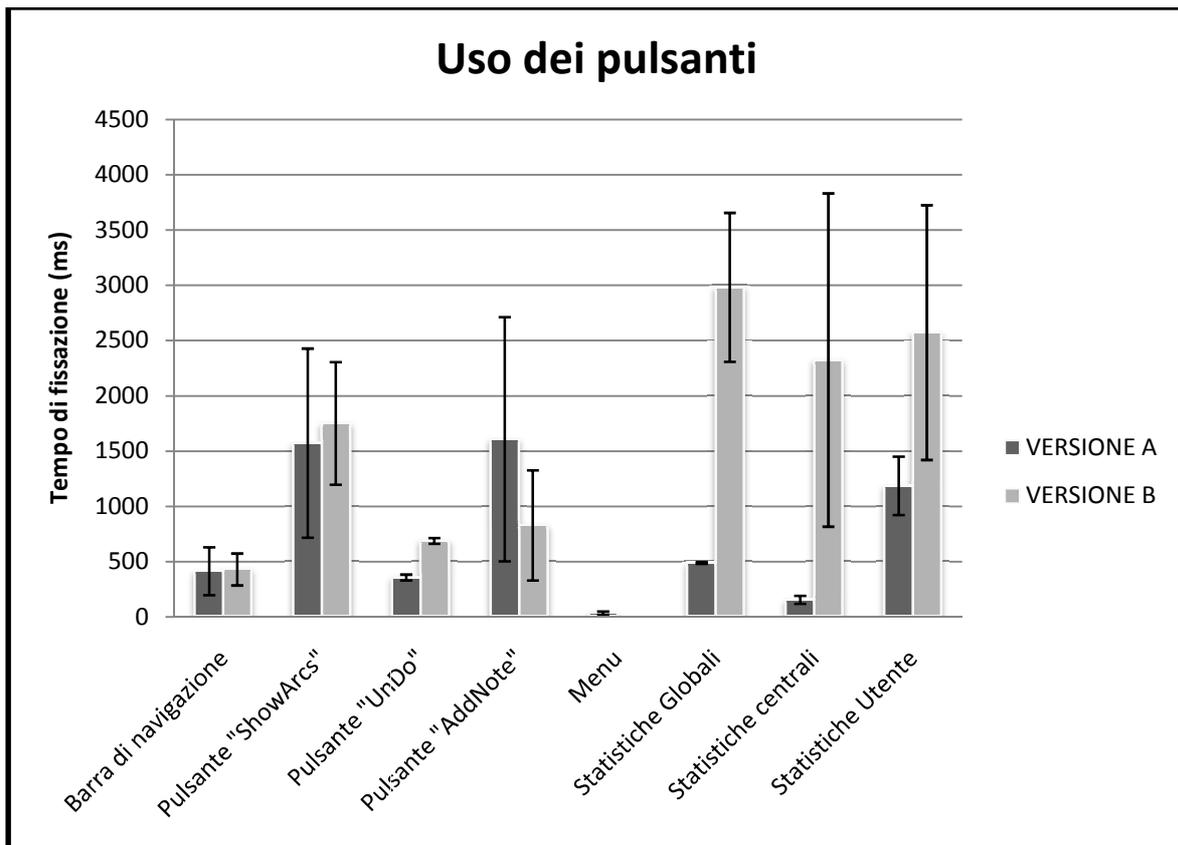
Guardando i grafici si può notare che a parità, circa, del tempo medio di fissazione per il pulsante Crea Meeting tra la versione A e la versione B, il numero di fissazioni risulta molto maggiore nella versione A. Questa condizione è verificata dal test ANOVA ($\sigma=0.0008$) per il tempo medio di fissazione, che risulta significativamente diverso e più del doppio rispetto alla versione A. Questo fatto indica una difficoltà di comprensione del tasto nella seconda versione probabilmente dato dalla nuova disposizione del tasto, che precedentemente si trovava subito al di sotto della barra di navigazione. Non si riscontrano altre differenze significative tra le due versioni.

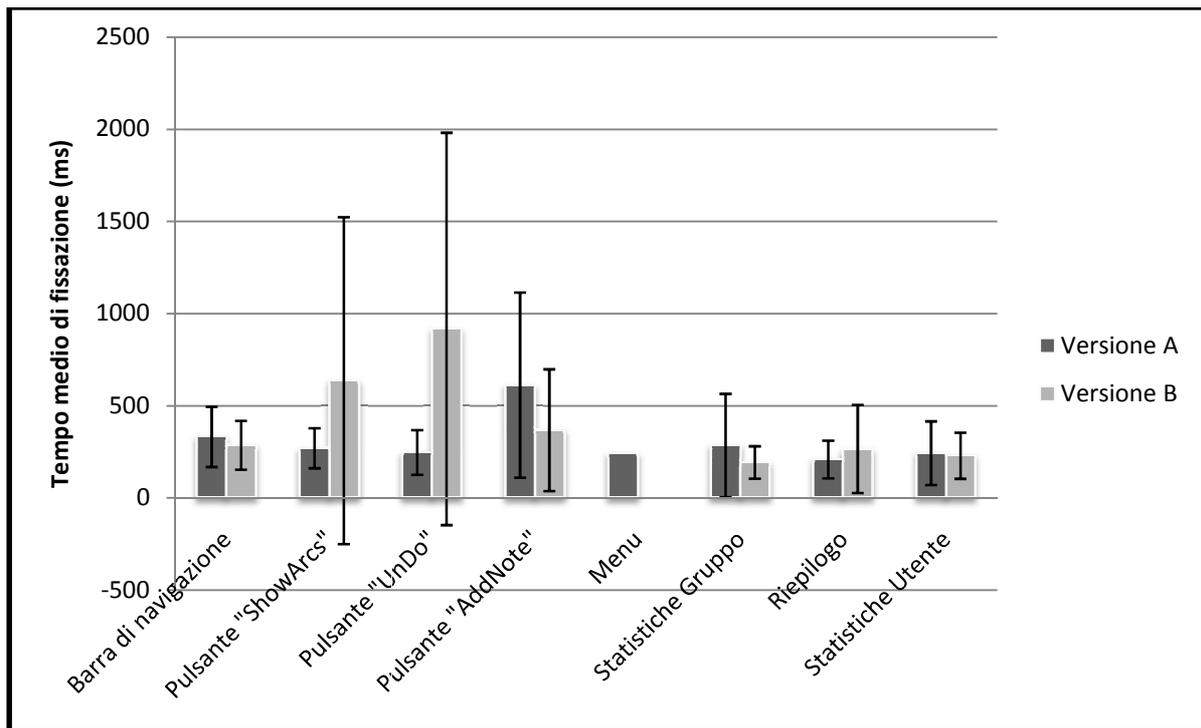




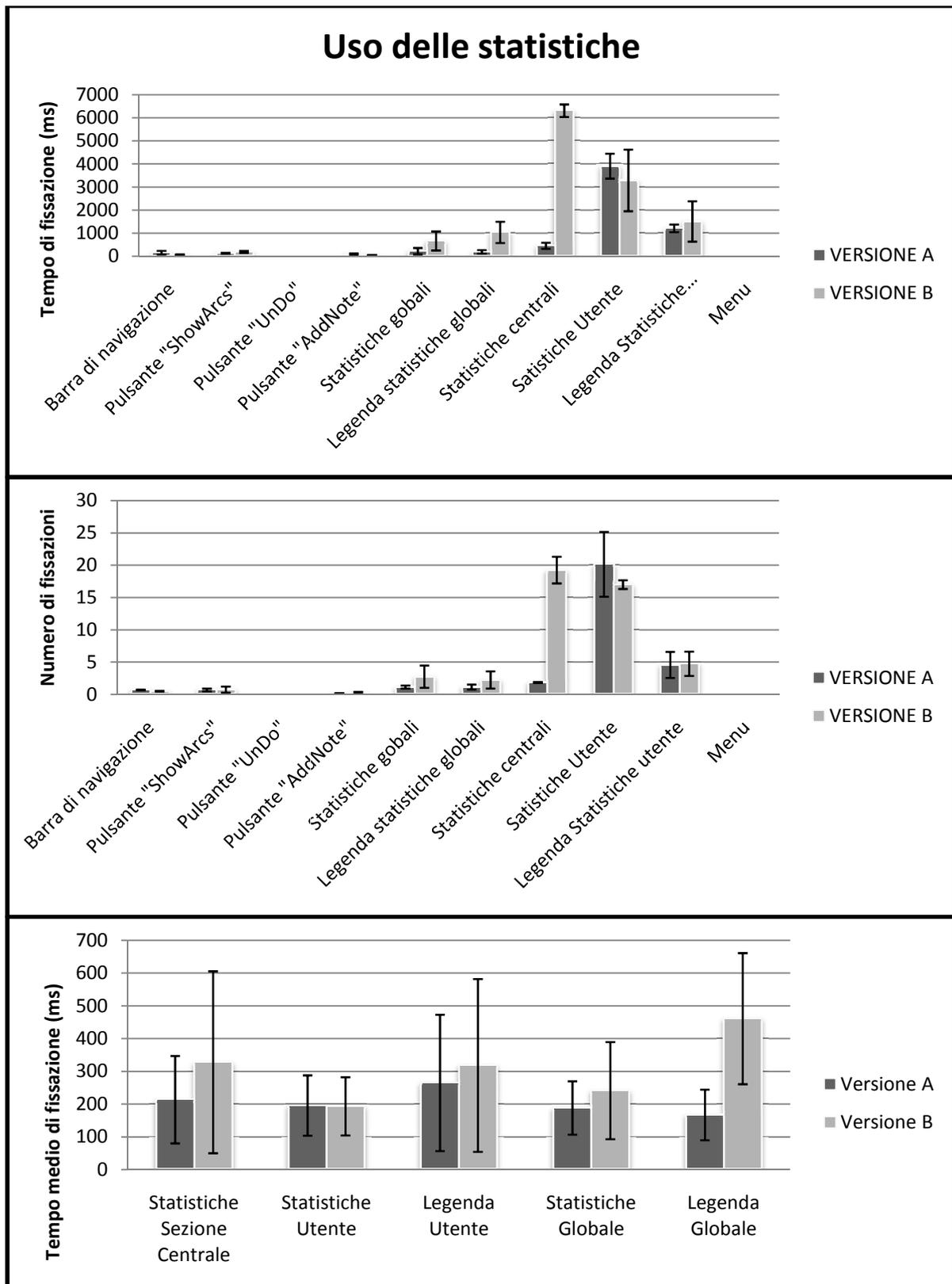
Nel task per l'avvio del meeting è presente una differenza significativa nei tempi di fissazione del pulsante per l'avvio del meeting (StartMeeting) che presenta una tempo medio di fissazione nella seconda versione rispetto alla prima ed un numero di fissazioni leggermente maggiore. Anche in questo caso la posizione della barra cambia la facilità di interpretazione del pulsante.



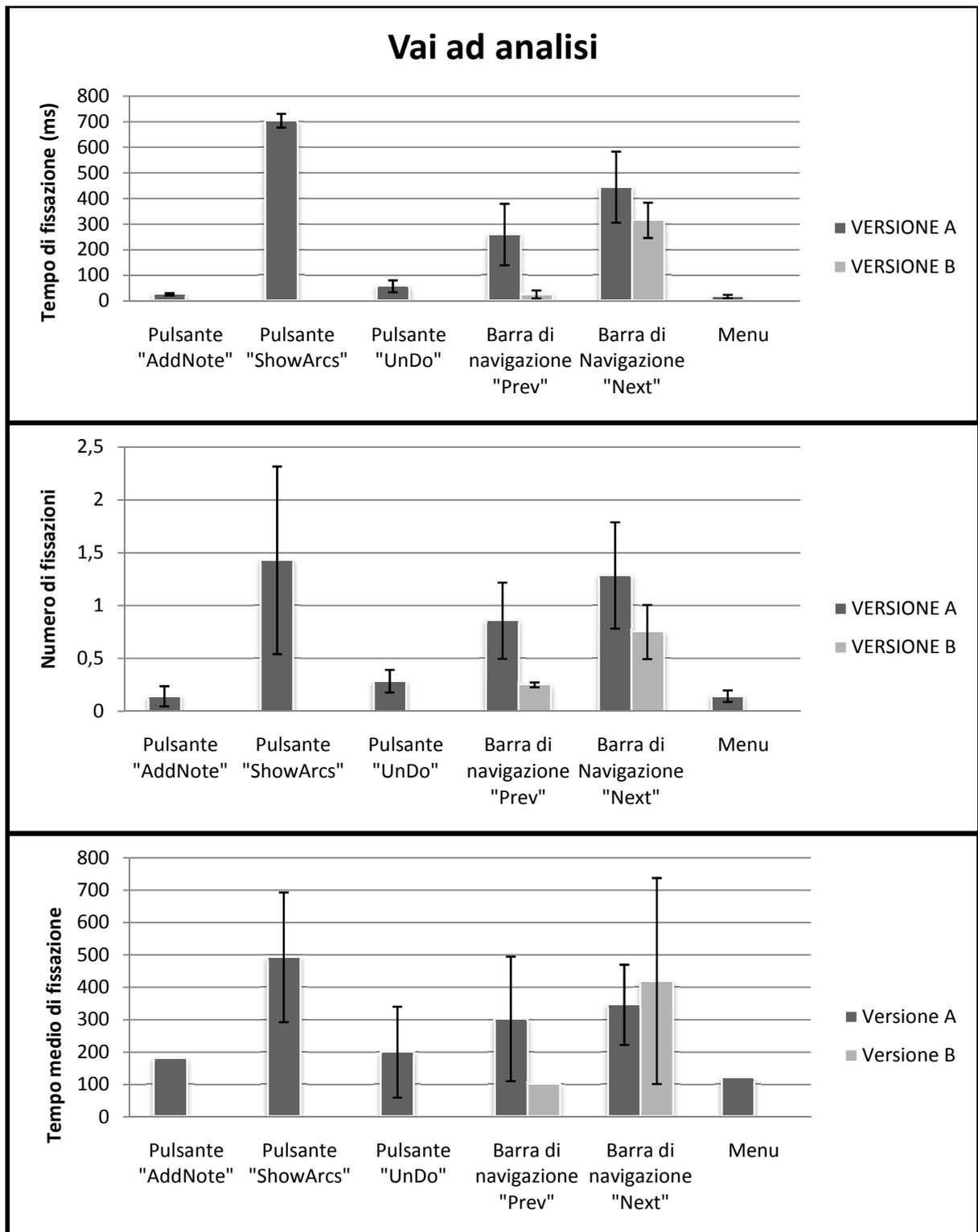




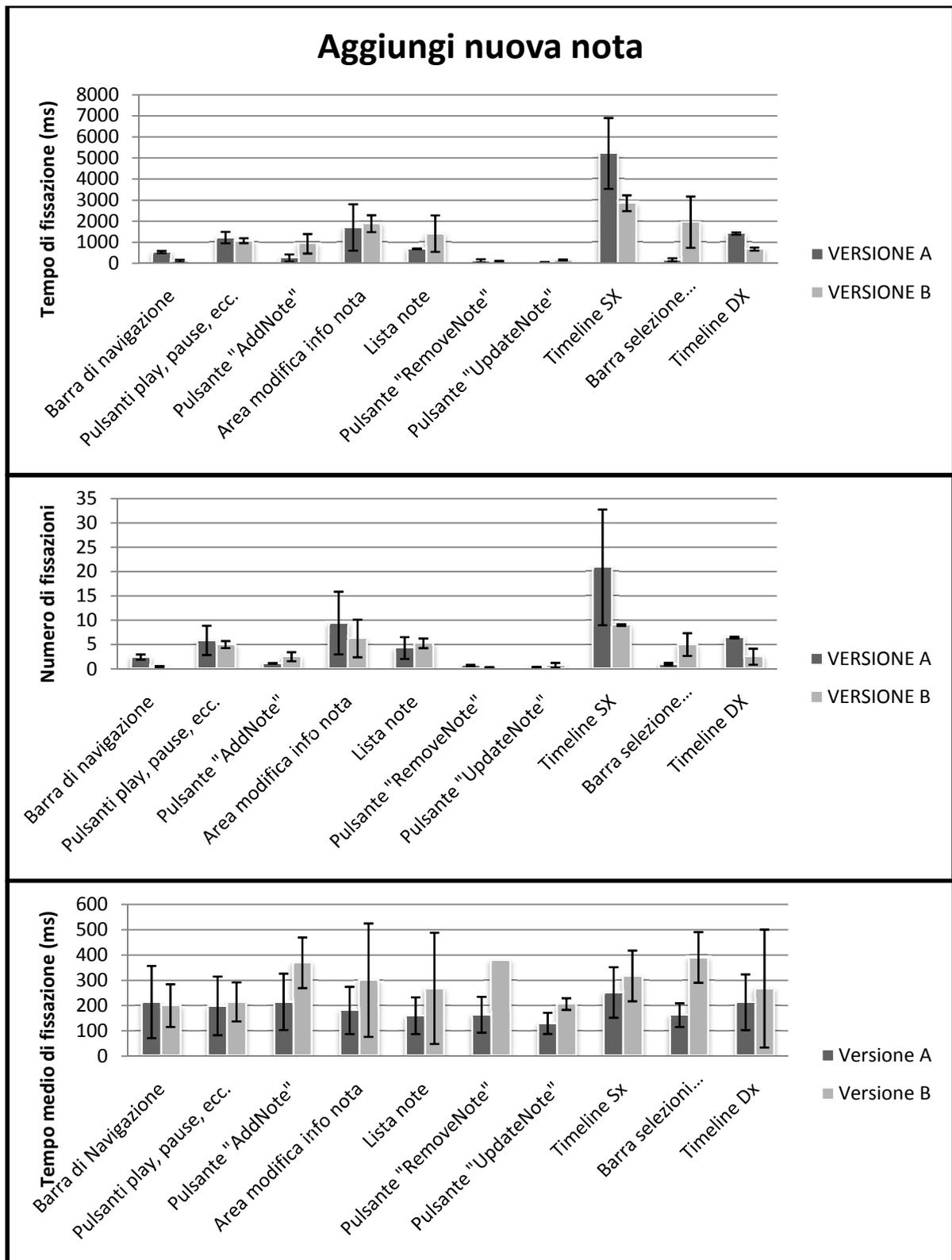
Nel caso dei pulsanti UnDo e ShowArcs si nota un tempo medio di fissazione significativamente diverso e molto più alto nella Versione B. Questo indica che la nuova posizione dei pulsanti non aiuta il riconoscimento dei due pulsanti. Differente è invece il caso del pulsante AddNote nel quale a parità di numero di fissazione il tempo totale di fissazioni risulta significativamente diverso e maggiore nel primo caso. Potrebbe, di conseguenza, essere opportuno separare i pulsanti UnDo e ShowArcs dal pulsante AddNote per migliorarne la learnability. Inoltre la diversa posizione dei pulsanti rende molto più visibile la sezione contenente le statistiche di gruppo (situata subito sotto nella versione B), e la sezione contenente gli indici principali, distogliendo l'attenzione dal task richiesto. Tale disposizione non è, perciò, ergonomicamente adatta al task.



Anche in questo caso, tra la versione A e la versione B, esiste una differenza significativa nel numero di fissazioni e nei tempi di fissazione, per quanto riguarda la sezione degli indici di rete. Anche in questo caso l'anomalia è da imputare alla differente disposizione dei tasti funzione (UnDo, AddNote, e ShowArcs), che, benchè in questo task, poco fissati, fungono da distruttore per l'intera operazione.

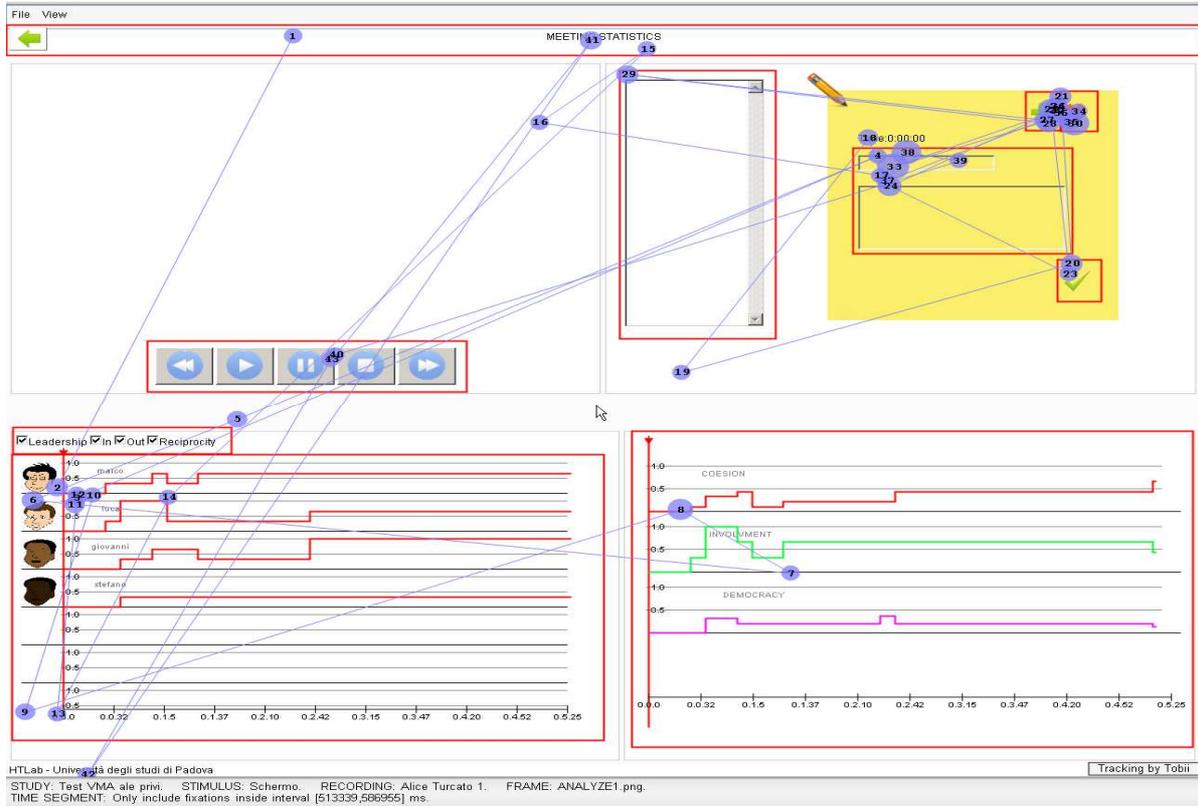


La diversa disposizione della barra di navigazione comporta un tempo medio di fissazione significativamente maggiore nella versione A del software, il che indica una difficoltà di interpretazione del comando. Un aumento anche del numero di fissazioni dei pulsanti funzione (UnDo, AddNote e ShowArcs) indica che il fatto è dovuto alla vicinanza di tali tasti vicino all'icona del pulsante "Previous" nella barra. L'utente tende, infatti a confondere il tasto di UnDo con il tasto per il ritorno alla schermata precedente a causa della somiglianza tra le due icone.

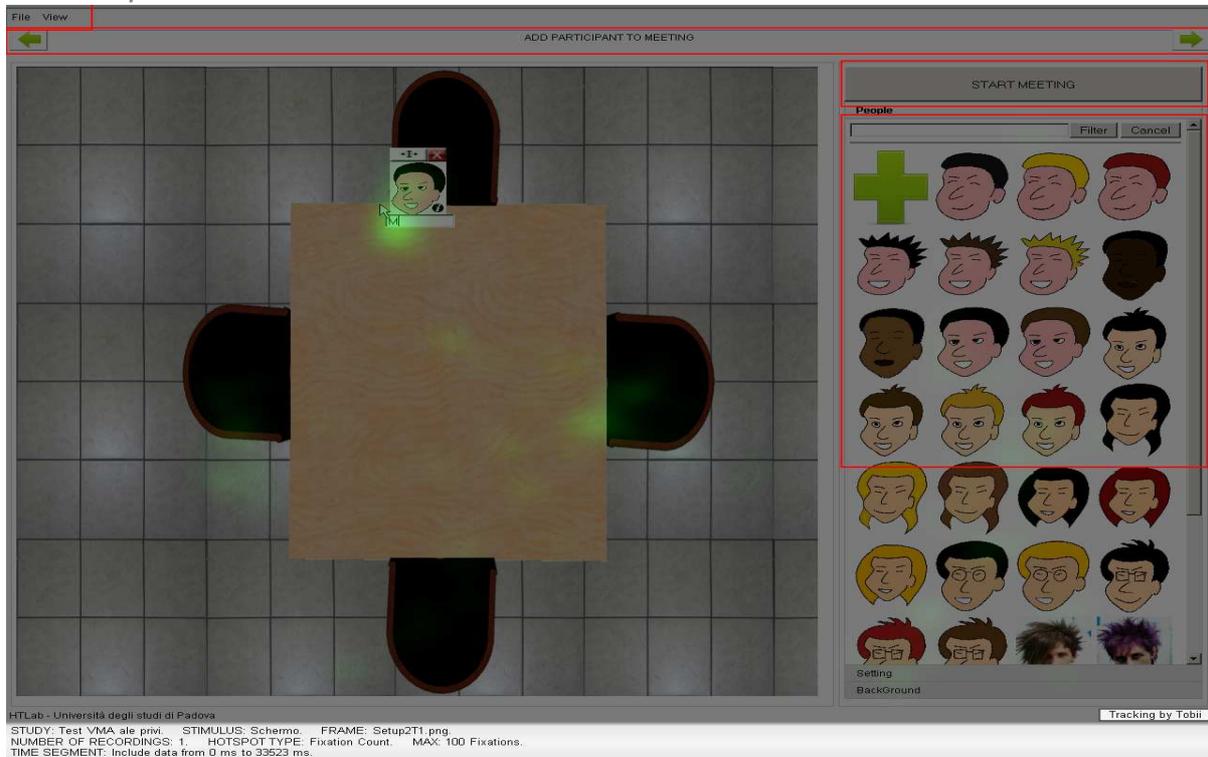


Il pulsante addNote presenta delle differenze significative nel tempo medio di fissazione e nel numero di fissazioni tra la versione A e la Versione B presentando una maggiore difficoltà cognitiva di interpretazione dell'icona in quest'ultima. Tale difficoltà è da imputarsi alla diversa icona presente nella seconda versione. La posizione risulta invece migliore, visti i risultati dei task che presentano una minore quantità di task non effettuati o errati nella seconda versione.

Sono qui riportate alcune schermate prodotte dal software per l'analisi dei movimenti oculari utilizzato (Tobii Clearview).



Visuale GazePoint



Visuale HotSpot

7 Conclusioni

In questo lavoro è stata presentata la progettazione, lo sviluppo e il testing ergonomico di un'applicazione per lo studio dei social network di riunioni. Si è mostrata la tecnologia utilizzata per lo sviluppo e le motivazioni perché tale piattaforma (la piattaforma LAMP) sia stata considerata la migliore. Si sono poi spiegate le funzionalità dell'applicazione e i possibili ambiti di utilizzo. Si è infine testato l'interfaccia tramite metodi di usabilità quali la checklist e l'analisi dei movimenti oculari.

Verranno qui di seguito illustrati brevemente alcuni possibili sviluppi di questo lavoro.

7.2 Website Analysis

L'interfaccia e gli algoritmi sviluppati e testati per questo lavoro si prestano per essere applicati anche ad altri ambiti di utilizzo.

Un esempio tipico è quello dell'analisi delle connessioni tra i siti web. Con opportune modifiche all'interfaccia e al motore dell'applicazione è possibile utilizzare lo stesso software il quale, inseriti un gruppo di domini web, tramite uno spider integrato ne analizza i link e ne ricava la matrice di adiacenza per il calcolo degli indici. Questo tipo di funzionalità può risultare utile a vari scopi. Si può innanzitutto studiare quale tra i vari siti considerati risulta il più centrale, oppure quale è poco collegato agli altri e quindi meno raggiungibile, per poter poi ottimizzare la rete di link. È inoltre possibile effettuare degli studi incrociati con dati di altro tipo come, ad esempio, la rete di relazioni tra gli amministratori delegati, o gli azionisti di un gruppo di aziende. È possibile confrontare le due reti per studiarne le somiglianze e le differenze.

Al momento è ancora in fase di sviluppo una versione del software, che effettua una analisi di questo tipo.

La problematica principale riguardante questo tipo di applicazione è quella dell'utilizzo di uno spider performante e abbastanza modulare da essere avviato in un server, e che ne utilizzi in maniera ottimale le risorse. Per la versione al momento in sviluppo è utilizzata la classe `urllib` di python per il reperimento delle pagine web e la libreria `SGMLlib` per il *parsing* delle pagine. Entrambe le librerie sono contenute nel set standard di Python. Questa scelta permette di creare degli spider semplici da utilizzare e leggeri dal punto di vista computazionale.

7.3 Nuove funzionalità dell'applicazione esistente

La struttura modulare del software si presta ad alcune aggiunte e modifiche:

1. Pagina setup: per la pagina di setup sono state pensate, già in fase di beta testing alcune modifiche al sistema di inserimento degli utenti; oltre al trascinamento delle icone (già presente), è stata progettata la possibilità di introdurre un sistema di creazione dell'avatar partendo da una serie grafiche predefinite (come colore degli occhi, dei capelli, forma del viso, della bocca, ecc.);
2. Struttura modulare: l'applicazione è creata lato server con una struttura modulare, sarà, perciò, possibile in futuro introdurre il calcolo di nuovi indici, per adattare le funzionalità dell'applicazione ad differenti tipologie di rete.
3. Export dei dati: attualmente l'applicazione esporta il solo listato delle comunicazioni e il report degli indici; per il futuro si introdurrà un sistema di reportistica più avanzato

8. Appendice

8.1 Test ANOVA sulla Checklist

ANOVA						
		Sum of Squares	df	Mean Square	F	Sig.
Affermazione 1	Between Groups	1,108	1	1,108	,575	,464
	Within Groups	21,200	11	1,927		
	Total	22,308	12			
Affermazione 2	Between Groups	1,402	1	1,402	1,128	,311
	Within Groups	13,675	11	1,243		
	Total	15,077	12			
Affermazione 3	Between Groups	1,731	1	1,731	2,004	,185
	Within Groups	9,500	11	,864		
	Total	11,231	12			
Affermazione 4	Between Groups	,433	1	,433	1,228	,291
	Within Groups	3,875	11	,352		
	Total	4,308	12			
Affermazione 5	Between Groups	,192	1	,192	,604	,453
	Within Groups	3,500	11	,318		
	Total	3,692	12			
Affermazione 6	Between Groups	2,633	1	2,633	2,118	,174
	Within Groups	13,675	11	1,243		
	Total	16,308	12			
Affermazione 7	Between Groups	,769	1	,769	1,058	,326
	Within Groups	8,000	11	,727		
	Total	8,769	12			
Affermazione 8	Between Groups	,031	1	,031	,037	,851
	Within Groups	9,200	11	,836		
	Total	9,231	12			
Affermazione 9	Between Groups	,769	1	,769	,604	,453
	Within Groups	14,000	11	1,273		
	Total	14,769	12			
Affermazione 10	Between Groups	4,431	1	4,431	5,538	,038
	Within Groups	8,800	11	,800		
	Total	13,231	12			
Affermazione 11	Between Groups	,556	1	,556	,482	,502
	Within Groups	12,675	11	1,152		
	Total	13,231	12			
Affermazione 12	Between Groups	,277	1	,277	,238	,635

	Within Groups	12,800	11	1,164		
	Total	13,077	12			
Affermazione 13	Between Groups	,017	1	,017	,034	,858
	Within Groups	5,675	11	,516		
	Total	5,692	12			
Affermazione 14	Between Groups	,233	1	,233	,628	,445
	Within Groups	4,075	11	,370		
	Total	4,308	12			
Affermazione 15	Between Groups	,325	1	,325	,370	,556
	Within Groups	9,675	11	,880		
	Total	10,000	12			
Affermazione 16	Between Groups	,433	1	,433	,343	,570
	Within Groups	13,875	11	1,261		
	Total	14,308	12			
Affermazione 17	Between Groups	,069	1	,069	,162	,695
	Within Groups	4,700	11	,427		
	Total	4,769	12			
Affermazione 18	Between Groups	,094	1	,094	,222	,647
	Within Groups	4,675	11	,425		
	Total	4,769	12			
Affermazione 19	Between Groups	,002	1	,002	,004	,950
	Within Groups	5,075	11	,461		
	Total	5,077	12			
Affermazione 20	Between Groups	,492	1	,492	,315	,586
	Within Groups	17,200	11	1,564		
	Total	17,692	12			
Affermazione 21	Between Groups	3,077	1	3,077	2,418	,148
	Within Groups	14,000	11	1,273		
	Total	17,077	12			
Affermazione 22	Between Groups	3,723	1	3,723	3,657	,082
	Within Groups	11,200	11	1,018		
	Total	14,923	12			
Affermazione 23	Between Groups	,277	1	,277	,448	,517
	Within Groups	6,800	11	,618		
	Total	7,077	12			
Affermazione 24	Between Groups	,031	1	,031	,026	,876
	Within Groups	13,200	11	1,200		
	Total	13,231	12			
Affermazione 25	Between Groups	1,508	1	1,508	1,121	,312
	Within Groups	14,800	11	1,345		
	Total	16,308	12			
Affermazione 26	Between Groups	1,202	1	1,202	3,412	,092

	Within Groups	3,875	11	,352		
	Total	5,077	12			
Affermazione 27	Between Groups	,008	1	,008	,008	,929
	Within Groups	10,300	11	,936		
	Total	10,308	12			
Affermazione 28	Between Groups	,467	1	,467	,417	,533
	Within Groups	11,200	10	1,120		
	Total	11,667	11			
Affermazione 29	Between Groups	1,402	1	1,402	1,594	,233
	Within Groups	9,675	11	,880		
	Total	11,077	12			
Affermazione 30	Between Groups	1,202	1	1,202	1,113	,314
	Within Groups	11,875	11	1,080		
	Total	13,077	12			
Affermazione 31	Between Groups	,123	1	,123	,091	,768
	Within Groups	14,800	11	1,345		
	Total	14,923	12			
Affermazione 32	Between Groups	,094	1	,094	,062	,808
	Within Groups	16,675	11	1,516		
	Total	16,769	12			
Affermazione 33	Between Groups	,017	1	,017	,025	,878
	Within Groups	7,675	11	,698		
	Total	7,692	12			
Affermazione 34	Between Groups	,048	1	,048	,077	,787
	Within Groups	6,875	11	,625		
	Total	6,923	12			
Affermazione 35	Between Groups	,048	1	,048	,077	,787
	Within Groups	6,875	11	,625		
	Total	6,923	12			

8.2 Test ANOVA sul questionario

ANOVA						
		Sum of Squares	df	Mean Square	F	Sig.
Coerenza secondo gli utenti delle informazioni ricavate sul Leader	Between Groups	,576	1	,576	,834	,388
	Within Groups	5,524	8	,690		
	Total	6,100	9			
Coerenza secondo gli utenti delle informazioni ricavate sulla persona che ha parlato meno	Between Groups	,686	1	,686	,960	,356
	Within Groups	5,714	8	,714		
	Total	6,400	9			
Coerenza secondo gli utenti delle informazioni ricavate sull'interazione tra i partecipanti	Between Groups	,305	1	,305	,595	,463
	Within Groups	4,095	8	,512		
	Total	4,400	9			
Comprensibilità secondo gli utenti delle informazioni	Between Groups	,386	1	,386	,400	,545
	Within Groups	7,714	8	,964		
	Total	8,100	9			
Utilità secondo gli utenti delle informazioni	Between Groups	,076	1	,076	,981	,0.03
	Within Groups	5,524	8	,690		
	Total	5,600	9			

8. Bibliografia

1. Rob Cross, Stephen P. Borgatti, Andrew Parker, 2002, *Making Invisible Work Visible: Using Social Network Analysis To Support Strategic Collaboration*, California Management Review Vol. 44, No. 2 Winter 2002
2. Stephen P. Borgatti, José Luis Molina, 2003, *Ethical and Strategic Issues in Organizational Social Network Analysis*, The Journal Of Applied Behavioral Science, Vol. 39 No. 3
3. Francesco Martino, Roberto Bau`, Anna Spagnolli, Luciano Gamberini, 2009, *Presence in the age of social networks: augmenting mediated environments with feedback on group activity*, Springer-Verlag London Limited
4. Joan Morris Dimicco, Walter Bender, 2004 *Second Messenger: Increasing The Visibility Of Minority Viewpoints With A Face-To-Face Collaboration Tool*, ACM 1-58113-815-6/04/0001
5. Joan Morris DiMicco, Katherine J. Hollenbach, Walter Bender, 2006, *Using Visualizations to Review a Group's Interaction Dynamics*, CHI 2006, ACM 1-59593-298-4/06/0004.
6. Dave Crane, Bear Bibeault, Jord Sonneveld, 2007, *Ajax in Practice*, Greenwich.
7. David Flanagan, 2000, *JavaScript - La guida*, Apogeo.
8. Alan Grosskurth, Michael W. Godfrey, David R. Cheriton, 2006, *Architecture and evolution of the modern web browser*,
<http://grosskurth.ca/papers/browser-archevol-20060619.pdf>
9. John Scott, 1997, *L'analisi delle reti sociali*, La nuova Italia scientifica
10. Mozilla Developer Network, 2010, *Canvas Tutorial, Basic Usage*,
https://developer.mozilla.org/en/Canvas_tutorial/Basic_usage

11. W3C Consortium 2010, *HTML5 (including next generation additions still in development) Draft Standard*, <http://www.whatwg.org/specs/web-apps/current-work/multipage/the-canvas-element.html#security-with-canvas-elements>
12. SciPy Foundation, 2010, *SciPy Getting Started*, <http://new.scipy.org/getting-started.html>
13. W3C Consortium, 2009, *Document Object Model (DOM)*, <http://www.w3.org/DOM/#what>
14. Ian Hickson, *HTML5: A vocabulary and associated APIs for HTML and XHTML, Editor's Draft*, <http://dev.w3.org/html5/spec/Overview.html#refsHTMLDIFF>
15. A. van Kesteren , 2010, *(Non-normative) HTML5 differences from HTML4*, <http://dev.w3.org/html5/html4-differences/>
16. James Reason, 1994, *L'errore Umano*, Il Mulino
17. Carpenter, Roger H.S., 1988, *Movements of the Eyes (2nd ed.)*, Pion Ltd
18. Matthew A. Russel, 2008, *Dojo The Definitive Guide*, O'Reilly
19. James E. Harmon, 2008, *Dojo: Using The Dojo Javascript Library to Build AJAX Applications*, Addison-Wesley
20. B.J. Fogg, Shawn Tseng, 1999, *Credibility and Computing Technology*, Communications of the ACM Volume 42, Issue 5
21. Jerry Grossman, 2009, *Facts about Erdős Numbers and the Collaboration Graph*, <http://www4.oakland.edu/enp/trivia/>

22. Alex Chaffee, 2000, *What is a web application (or "webapp")*,
<http://www.jguru.com/faq/view.jsp?EID=129328>

23. Mazzone, E., 2002, *La Social Network Analysis a supporto delle interazioni nelle comunità virtuali per la costruzione di conoscenza*, TD, Vol. 35, No. 2, pp. 54-63