

ALMA MATER STUDIORUM · UNIVERSITÀ DI
BOLOGNA

SCUOLA DI SCIENZE
Corso di Laurea in Informatica

**Analisi del comportamento degli
utenti nell'utilizzo di Editor
collaborativi per documenti di testo**

Relatore:
Dottor Angelo Di Iorio

Presentata da:
Chiara Zaccarelli

Sessione
Anno Accademico 2015/2016

*Alla mia famiglia, a Matteo
e a tutti coloro che mi hanno sostenuto
in questi ultimi anni*

Introduzione

Gli Editor collaborativi in Real Time o, in breve, RTCE come Google Docs e Etherpad Lite sono, al giorno d'oggi, strumenti largamente diffusi e sempre più persone, me compresa, lo utilizzano per i propri progetti sul lavoro, nello studio e nella vita privata.

Si tratta di vere e proprie applicazioni che permettono di scrivere e modificare testo e, allo stesso tempo, fanno in modo che più individui possano utilizzare lo stesso documento nello stesso momento, osservando ciò che fanno gli altri e agendo di conseguenza.

È da una ricerca svolta su questi Editor, che tenta di determinare come avviene la scrittura collaborativa nello spazio e nel tempo, che nasce il bisogno a cui si tenterà di dare una soluzione in questa esposizione.

Ciò che si vuole fare è riuscire a determinare un modo con cui poter registrare le azioni che un utente compie sull'interfaccia durante l'uso di questi strumenti.

A questo scopo si è deciso di implementare uno strumento alternativo utilizzando come base Etherpad Lite, un servizio RTCE Open Source interamente sviluppato in Javascript, che permetta di immagazzinare tali dati.

In aggiunta si è pensato di ideare anche un secondo strumento, che recuperi i dati immagazzinati e li rielabori in modo da poterli esporre in un modo strutturato e di facile comprensione, ossia dividendoli tra Autori, Pad (l'ambiente in cui i singoli documenti vengono visualizzati e modificati), sessioni e tipi di dato, il tutto con l'aiuto di tabelle e grafici.

Questi strumenti hanno però grandi potenzialità e attraverso alcune piccole migliorie potrebbero risultare ancora più performanti e utili.

Nel primo capitolo si parlerà in generale di cosa sono gli Editor Collaborativi e di quali studi sono già stati affrontati riguardanti questo argomento. Nel secondo capitolo verrà introdotto il problema, ossia la necessità di registrare le azioni degli utenti, e presentata la soluzione da me implementata, UICEM. Nel capitolo tre saranno esaminati tutti i dettagli implementativi, mentre nel capitolo quattro verrà svolta un'analisi dello strumento e delle migliorie che potrebbero essere implementate per renderlo ancora più efficiente.

Indice

Introduzione	i
1 Real Time Collaborative Editing: stato dell'arte	1
1.1 Real Time collaborative editing	1
1.2 Panoramica sugli studi già condotti	2
1.2.1 Studi sulla collaborazione e sull'apprendimento	2
1.2.2 Studi sulla concorrenza	4
2 UICEM: uno strumento per monitorare il comportamento degli utenti nell'uso di RTCE	7
2.1 Problema: caratterizzazione del comportamento degli utenti	7
2.1.1 Un caso studio: Etherpad Lite	9
2.2 UICEM	9
2.2.1 Tipi di comportamento visualizzato	10
2.2.2 Etherpad Lite M	14
2.2.3 UICEM Display	19
3 Dettagli implementativi	27
3.1 Struttura del Database	27
3.2 Etherpad Lite M	31
3.3 UICEM Display	32
3.3.1 Recupero e Organizzazione dei Dati	33
3.3.2 Sessioni	35
3.3.3 Bookmarks	36

3.3.4	Grafici	38
4	Valutazione	39
4.1	Test	39
4.1.1	Criticità	43
4.2	Miglioramenti	43
4.2.1	Filtri di Ricerca	43
4.2.2	Integrazione tra meccanismi di editing e interfaccia . . .	44
4.2.3	Confronto tra due o più elementi dello stesso tipo . . .	44
4.2.4	Estensione ad altri Editor	44
4.3	Usabilità e Mantenimento del Codice	45
	Conclusioni	47
	Bibliografia	49

Capitolo 1

Real Time Collaborative Editing: stato dell'arte

La sigla RTCE sta per Real Time Collaborative Editor, ossia Editor collaborativo in tempo reale. Si tratta di un genere di programma che permette a più Autori di poter lavorare su uno stesso documento anche contemporaneamente.

Si tratta di strumenti che, oltre ad avere un'indubbia utilità, destano particolare interesse per le proprie peculiari caratteristiche, motivo per cui sono stati utilizzati come oggetto di studio di diverse ricerche nel corso degli ultimi anni.

1.1 Real Time collaborative editing

Nonostante il concetto di Editor Collaborativo fosse già stato teorizzato e dimostrato da Douglas Engelbart nel suo “The Mother of All Demos” nel 1968[7], gli Editor Collaborativi come li conosciamo oggi sono venuti alla luce solo grazie all'avvento del Web 2.0, diventando quella che è ormai una realtà ben consolidata il cui utilizzo risulta piuttosto diffuso[8].

Gli Editor Collaborativi utilizzano, in generale, una struttura come quella di editor di testo più “classici”, per esempio il Word di Windows o il

Writer di OpenOffice, alla quale viene aggiunta l'opportunità di poter far collaborare due o più persone in modo che possano scrivere e modificare uno stesso documento nello stesso momento, facendo particolare attenzione alla concorrenza tra revisioni simultanee. In alcuni casi vengono implementati anche servizi non prettamente inerenti alla scrittura del documento, ma che possono risultare utilissimi a livello comunicativo e organizzativo, come ad esempio una chat o una cronologia delle revisioni.

Tra gli Editor Collaborativi attualmente più conosciuti e utilizzati si trova senza dubbio Google Docs (parte del servizio di Cloud "Google Drive" offerto da Google), ma negli ultimi anni si sono diffusi numerosissimi servizi simili, anche di carattere Open Source. Tra questi ultimi spicca certamente Etherpad Lite.

L'utilizzo di questo tipo di applicativo si è diffuso sempre di più nel corso degli ultimi anni, grazie anche ai servizi di Cloud di cui spesso fanno parte, e possono essere utilizzati sia in ambito professionale e accademico, per esempio se si necessita di scrivere un articolo ma non si ha la possibilità di trovarsi di persona, che nel privato, grazie alla loro estrema immediatezza.

1.2 Panoramica sugli studi già condotti

Viste le loro particolarità, gli Editor Collaborativi risultano strumenti particolarmente interessanti da studiare e nel corso degli anni sempre più ricercatori hanno iniziato a utilizzarli come oggetto delle proprie analisi.

1.2.1 Studi sulla collaborazione e sull'apprendimento

Buona parte di queste ricerche si sofferma in modo particolare nel determinare in che modo queste applicazioni possano agire sull'apprendimento o sulla socialità tra i collaboratori.

Sono molte le domande a cui queste ricerche tentano di rispondere.

- In che modo avviene la collaborazione?

- Quali sono i punti di forza e quali quelli da migliorare?
- Nel caso ci fosse la possibilità, perché un autore dovrebbe voler scegliere di utilizzare un prodotto di questo tipo quando può trovarsi insieme al proprio partner di lavoro davanti alla stessa scrivania e lavorare faccia a faccia?
- In che modo possono aiutare o danneggiare il rapporto tra due persone?
- Possono avere effetti sull'apprendimento?

Una delle prime ricerche riguardanti questo ambito è stata quella di Chu e Kennedy che nel 2011[5] organizzarono un esperimento con alcuni studenti dell'Università di Hong Kong allo scopo di capire se strumenti come Media-Wiki o Google Docs risultassero utili per l'apprendimento collaborativo o meno, con risultati per lo più soddisfacenti.

Un altro studio di Birnholtz e Ibara[2] analizza il modo in cui un edit sul documento può essere percepito.

Un utente, infatti, può avvertire una modifica fatta da un collaboratore in modi molto diversi che possono esulare dall'ambito tecnico o task-oriented. Per esempio può essere avvertito come segno di interesse nei confronti del lavoro che si sta svolgendo o, al contrario, come un'azione poco rispettosa del lavoro già compiuto da un'altra persona. Sorgono, dunque, sia il problema dell'utente di capire quali fossero le intenzioni di chi ha editato, ma allo stesso tempo sta all'utente stesso cercare di trovare il modo migliore per editare il documento senza che le proprie modifiche vengano percepite come troppo invasive o poco rispettose da parte dei collaboratori. Come accade anche in molti altri campi[6], la "visibilità" delle modifiche gioca un ruolo molto importante, poiché ogni autore può osservare con chiarezza quali edit al testo sono stati fatti e quale sia stato il vero contributo di ogni suo collaboratore.

Brodhal e Hansen[4] hanno condotto, poi, un caso studio su alcuni gruppi di studenti assegnando loro la scrittura collettiva di un testo con il tassativo utilizzo di tecnologie RTCE allo scopo di capire quale fosse la percezione che

le persone hanno di strumenti di questo tipo in base alla propria età, al genere e alle esperienze pregresse in ambito informatico. In questo caso e in quelli precedenti i dati sono stati raccolti principalmente attraverso questionari, riflessioni scritte dei partecipanti o interviste. Si tratta di un approccio nel quale i dati raccolti risultano molto soggettivi, in quanto si analizza ciò che un individuo personalmente è riuscito ad osservare, provare e raccontare e ciò non sempre risulta assolutamente affidabile. In casi come questi, però, questo tipo di approccio risulta assolutamente efficace, anche e soprattutto perché il vero soggetto delle ricerche non è l'editor, bensì l'utente stesso. Si cerca di capire cosa l'utente pensa, cosa percepisce, quali sono i suoi problemi, non cosa effettivamente svolge mentre utilizza il servizio.

L'unica ricerca di questo ambito che, fino ad ora, ha utilizzato anche un altro tipo di fonte per la raccolta dei dati è quella di Birnholtz, Steinhardt e Pavese del 2013[3], con lo scopo di esaminare le impressioni che un utente ha del suo collaboratore anche in base al tipo di edit sul documento. A questo scopo, oltre ai questionari a cui gli studenti coinvolti hanno risposto dopo l'esperimento, è stata fatta anche una sistematica raccolta e analisi dei log: questi log vengono divisi in diversi gruppi, in particolare quelli riguardanti la comunicazione tra gli utenti (provenienti sia dalla chat integrata su Google Drive sia da modifiche temporanee nel documento stesso) e gli edits veri e propri. Anche in questo caso, però, il focus della ricerca è più sugli effetti sociali che gli editor collaborativi comportano, che su una precisa analisi di come e quando le funzionalità siano utilizzate.

1.2.2 Studi sulla concorrenza

Un secondo tipo di studi sugli Editor Collaborativi che sono stati affrontati fino ad ora sono quelli sulla concorrenza, ossia il modo in cui l'Editor gestisce i conflitti tra modifiche fatte nello stesso istante.

La ricerca di Ahmed-Nacer e colleghi del 2011[1] chiamato "Evaluating CRDTs for real-time document editing" si è proposta infatti di analizzare e confrontare tutti quelli che erano gli algoritmi più utilizzati dagli Editor

Collaborativi[9] allo scopo di valutare quali fossero quelli più efficienti. Per fare ciò l'Editor "Team Edit" è stato modificato in modo da implementare i diversi algoritmi da studiare e registrare alcune informazioni aggiuntive che potessero aiutare a calcolare informazioni come ad esempio il tempo che una determinata modifica impiega a propagarsi a tutti gli utenti.

Questo studio non è certamente l'unico che si ripropone di studiare le performance degli algoritmi di concorrenza per quanto riguarda gli Editor di testo, ma risulta particolarmente interessante come esempio da osservare per quanto riguarda l'approccio con cui si sono raccolti i dati attraverso log del programma stesso.

Indagini di questo tipo, però, sono molto distanti e affrontate a livello troppo basso rispetto a quella che si vuole affrontare in questa dissertazione.

Capitolo 2

UICEM: uno strumento per monitore il comportamento degli utenti nell'uso di RTCE

2.1 Problema: caratterizzazione del comportamento degli utenti

A differenza di quanto già esposto precedentemente, il problema che questa esposizione tenta di risolvere è quello di poter monitorare quanto più accuratamente possibile le azioni che un utente compie su un editor durante una sessione di scrittura collaborativa.

Ciò che si intende studiare si pone al centro dei tipi di ricerche descritti sopra poiché si tenta di studiare un aspetto degli Editor più ad alto livello rispetto alle analisi sugli algoritmi, ma più tecnico rispetto alla possibilità di capire quali effetti abbiano questi strumenti sulla socialità, in altri termini quando e come avviene la scrittura collaborativa.

Si tratta, dunque, di riuscire a determinare come i vari Autori utilizzino questi strumenti, di come aggiungano le proprie modifiche al documento e come si comportino sull'interfaccia in base agli strumenti che hanno a dispo-

sizione. Insomma, di come la scrittura collaborativa avviene nello spazio e nel tempo.

Per fare questo, tuttavia, ciò che gli Editor Collaborativi nativamente offrono non è sufficiente.

Che si tratti di Etherpad Lite, Google Docs o un qualsiasi altro servizio RTCE, ciò che viene immagazzinato sul Database è per lo più relativo alle funzionalità di mantenimento del documento: vengono salvate puntualmente tutte le modifiche, associandole all'autore e al momento in cui sono state fatte, vengono salvati i dati relativi ad ogni autore e, se presenti, vengono salvati anche i messaggi della chat e così via.

Ci sono però elementi che risulterebbero altrettanto interessanti da studiare, ma che, al contrario, non risultano necessari per l'editor da immagazzinare poiché, in condizioni normali, risulterebbero superflui. Si tratta delle azioni che l'Autore compie sull'interfaccia. Per la gestione di un documento, infatti, non è necessario sapere quante volte un Autore ha aperto e chiuso la chat o quante volte abbia cambiato il proprio nome utente. Queste informazioni, però, possono essere interessanti da studiare. L'unico modo per poter recuperare informazioni di questo genere, visto che l'editor non fornisce i dati precisi su cui lavorare, è quello di predisporre il programma stesso a salvare sistematicamente i dati ogni volta che l'utente interviene sull'interfaccia. La maggior parte delle informazioni necessarie a questo scopo, infatti, non è catturabile se non dall'interno del programma.

L'idea è quella di individuare le operazioni che l'Autore può compiere sull'interfaccia e salvarle sul database, facendo attenzione a salvare anche il momento in cui vengono compiute, da chi vengono svolte e su quale documento.

Non solo: si è pensato di aggiungere uno strumento che permetta ai ricercatori di analizzare ancora meglio i dati, raggruppandoli, smistandoli ed esponendoli utilizzando dei grafici.

2.1.1 Un caso studio: Etherpad Lite

Etherpad Lite¹ è un progetto scritto utilizzando il linguaggio Javascript e il suo ambiente di sviluppo Node.js², nato allo scopo di riscrivere con tecnologie più moderne uno dei primi progetti di editor collaborativi, un'applicazione indipendente chiamata Etherpad, nata nel 2008 e chiusa dopo soli due anni in seguito all'acquisizione da parte di Google. Etherpad Lite risulta però molto più leggero del suo predecessore e, come già anticipato, il suo codice è completamente Open Source.

L'alta personalizzazione e la natura Open Source di Etherpad Lite fanno sì che possa essere considerato lo strumento migliore da utilizzare per il tipo di lavoro precedentemente esposto.

2.2 UICEM

Una soluzione al problema già presentato può essere resa disponibile grazie a UICEM (User Interface Collaborative Editing Monitor), un tool da me sviluppato. In realtà non si tratta di un software unitario che svolge tutte le operazioni necessarie, bensì di un progetto diviso in due moduli distinti e autonomi che lavorano in fasi diverse per raggiungere il medesimo scopo.

Il primo modulo ha il compito di acquisire i dati: si tratta infatti di una versione di Etherpad Lite modificata chiamata Etherpad Lite M (Monitoring), che permette di salvare sul database non solo i dati necessari per il mantenimento dei documenti e degli utenti, bensì anche alcune informazioni aggiuntive necessarie per lo studio del comportamento dell'utente sull'interfaccia, tra cui interazione con i bottoni, con i menù, con la chat, dati sulle sessioni e movimenti di scrolling. L'utente che dovrà utilizzarlo (in questo caso colui che partecipa al test) non ha alcun modo per capire se quella su cui lavorerà sia la versione classica oppure quella da me elaborata.

¹<http://etherpad.org/>

²<https://nodejs.org/en/>

Il secondo modulo (UICEM Display) avrà lo scopo di recuperare ed esporre alcune delle informazioni immagazzinate a seguito di un esperimento, in modo che risultino facilmente fruibili e interpretabili. In questo caso ad utilizzare il Display saranno gli stessi ricercatori che intendono studiare i dati.

L'unica connessione che avviene tra l'uno e l'altro modulo è una condivisione del database: il modulo di esposizione dei dati, infatti, può essere utilizzato contemporaneamente in più istanze anche su database MySQL differenti, a patto che se ne conoscano i dati d'accesso.

2.2.1 Tipi di comportamento visualizzato

Nonostante i tipi di dati raccolti da Etherpad Lite M siano molti, ne sono stati selezionati solo alcuni da far visualizzare a UICEM Display, ossia quelli che potrebbero risultare più interessanti al fine dello studio.

Cambio di nome (“Name Change”)

In primo luogo si vuole osservare quante volte e come l'Autore cambia il proprio username, ossia ogni volta che l'Autore apre il menù degli Autori collegati all'interno del Pad e interagisce con la casella di input di testo contenente il suo nome.

Etherpad Lite M registra sia il vecchio nome che il nuovo, in modo da poter costruire una cronologia accurata dei momenti in cui l'autore ha deciso di cambiare il proprio nome visualizzato.

Ogni revisione contiene un solo cambio di nome che verrà visualizzato nell'applicazione con questo aspetto:

Username of user [identificativo univoco utente] changed from [vecchio nome] to [nuovo nome]
[timestamp]

Per esempio:

Username of user a.YP0tCfsiuoKOj02 changed from Chiara to Chiara2
Wed Jul 13 2016 13:04:23 GMT+0200 (ora legale Europa occidentale)

Tutte le revisioni di questo tipo saranno identificate con l’etichetta “Name Change”.

Chat

Il secondo comportamento da osservare riguarda le modalità di utilizzo della chat. Nel database sono presenti molte informazioni diverse: quando e in che modo viene aperta la chat, quando viene chiusa, quando vengono inviati dei messaggi e quando l’utente scrolla all’interno della chat per recuperare messaggi vecchi. Tuttavia nello strumento di visualizzazione si è scelto per semplicità di osservare solamente quando la chat viene aperta, quando viene chiusa e quando viene eseguito uno scroll e ogni log all’interno di UICEM Display ha questo aspetto:

```
User [identificativo univoco utente]: chat is now [open/closed/scrolling]
                                     [timestamp]
```

Per esempio:

```
User a.3Y0Utg9rfvnMpSGf: chat is now open
Wed Jul 13 2016 13:04:23 GMT+0200 (ora legale Europa occidentale)
```

Le revisioni di questo tipo sono identificate, semplicemente, con l’etichetta “Chat”.

Annulla e Ripeti (“Undo/Redo”)

Il terzo comportamento riguarda l’utilizzo delle funzioni di “annulla” e “ripeti”. Ogni volta che l’Autore decide di cliccare sul bottone “annulla” o sul bottone “ripeti”, tale comportamento viene salvato sul database.

I log nello strumento sono visualizzati in questo modo:

User [identificativo univoco utente] clicked button: [undo/redo]
[timestamp]

Per esempio:

User a.YP0tCfsiuroKOj02 clicked button: undo
Wed Jul 13 2016 13:04:23 GMT+0200 (ora legale Europa occidentale)

Le revisioni di questo tipo vengono identificate con l'etichetta "Undo/Redo".

Impostazioni di Visualizzazione ("Settings")

Il quarto comportamento determina quante volte l'autore utilizza il menù di opzioni di visualizzazione e le sue funzioni. Sul database vengono salvati log riguardanti:

1. i momenti in cui tale menù viene aperto o chiuso;
2. quando viene selezionata o deselezionata un'opzione al suo interno;
3. quando si fa una scelta sui due menù a tendina del cambio di lingua e cambio di font.

Nello strumento, i log vengono visualizzati in due modi diversi. Se il log si riferisce all'apertura o alla chiusura del menù è visualizzato in questo modo:

User [identificativo univoco utente] [opened/closed] settings menù
[timestamp]

Al contrario, se si tratta del cambiamento di una delle opzioni all'interno del menù è visualizzato così:

User [identificativo univoco utente] changed his settings:[tipo di modifica
fatta]
[timestamp]

Per esempio:

User a.YP0tCfsiuroKOj02 opened settings menù
Wed Jul 13 2016 13:04:23 GMT+0200 (ora legale Europa occidentale)
User a.3Y0Utg9rfvnMpSGf changed his settings: new font serif
Wed Jul 13 2016 13:04:23 GMT+0200 (ora legale Europa occidentale)

Le revisioni di questo tipo vengono identificate con l’etichetta “Settings”.

Scrolling

In ogni Editor di documenti che si rispetti, collaborativo o non che sia, se un documento è troppo lungo per essere visualizzato in una pagina sola, deve essere possibile poter fare scrolling lungo le sue pagine, in modo da poter permettere a ogni utente di leggere, osservare o anche modificare ogni suo paragrafo.

Le operazioni di scrolling possono risultare molto interessanti durante un’analisi dei comportamenti degli utenti, in quanto permettono di osservare in modo preciso come un Autore si muove quando è connesso al Pad e per questo motivo dette operazioni vengono a loro volta monitorate. Ogni volta che l’Autore si muove all’interno del documento, sul database viene registrato il momento in cui c’è stato il movimento insieme alla prima riga correntemente visualizzata.

Tuttavia quest’ultimo comportamento è certamente quello più complesso da studiare, poiché non è importante sapere quante volte l’autore ha eseguito uno scroll, quanto piuttosto sapere quando lo ha fatto e, in generale, come si muove all’interno del documento durante la sessione. Proprio per questo il numero di revisioni che vengono registrate riguardo questo fenomeno risulta poco significativo e sarà possibile analizzare correttamente questo comportamento solo entrando nel dettaglio di ogni singola sessione.

I log riguardanti lo scrolling sono strutturati in questo modo:

User [identificativo univoco utente] scrolled [movimento]
[timestamp]

In cui il movimento descrive come si muove l’utente. Per esempio:

User a.YP0tCfsiuoKOj02 scrolled down from line 1 to line 3
Wed Jul 13 2016 13:04:23 GMT+0200 (ora legale Europa occidentale)

Le revisioni di questo tipo vengono identificate con l'etichetta "Scrolling".

2.2.2 Etherpad Lite M

Come già illustrato, il primo modulo non è altri che una versione modificata di Etherpad Lite denominata Etherpad Lite M, nel quale M sta per Monitoring, che permette di memorizzare attraverso log sul database dettagli della sessione che non sarebbe possibile catturare altrimenti. Essendo questa una funzionalità che non ha bisogno di elementi aggiuntivi all'interno dell'interfaccia utente, l'interfaccia di questa nuova versione non differisce in nessun elemento da quella dell'originale Etherpad Lite. Questa caratteristica risulta fondamentale se lo scopo è quello di raccogliere dei dati più genuini possibili riguardanti l'esperienza dell'utente nei confronti dell'applicazione originale. Per questo motivo, in questa sezione mi limiterò a esporre quali siano le caratteristiche riguardanti l'interfaccia di una qualsiasi versione di Etherpad Lite e quali siano invece gli elementi che si intendono catturare.

Struttura generale del Pad e gestione degli Autori e dei loro edit

La prima azione da fare per lavorare con UICEM è accedere ad un Pad, nuovo o già utilizzato che sia. Una volta fatto la prima cosa che ci si trova davanti è l'editor di testo.

Iniziando a scrivere, il testo, di default, apparirà evidenziato in uno specifico colore: si tratta del colore associato all'Autore. Ogni Autore ha un colore diverso che permette di rendere perfettamente riconoscibili i punti in cui un determinato Autore abbia fatto un inserimento o una modifica. È possibile associare ogni Autore al proprio colore dal menù degli Autori collegati.

Il menù degli Autori collegati può essere aperto cliccando il bottone più a destra nella barra degli strumenti. Questo bottone, del colore predefinito

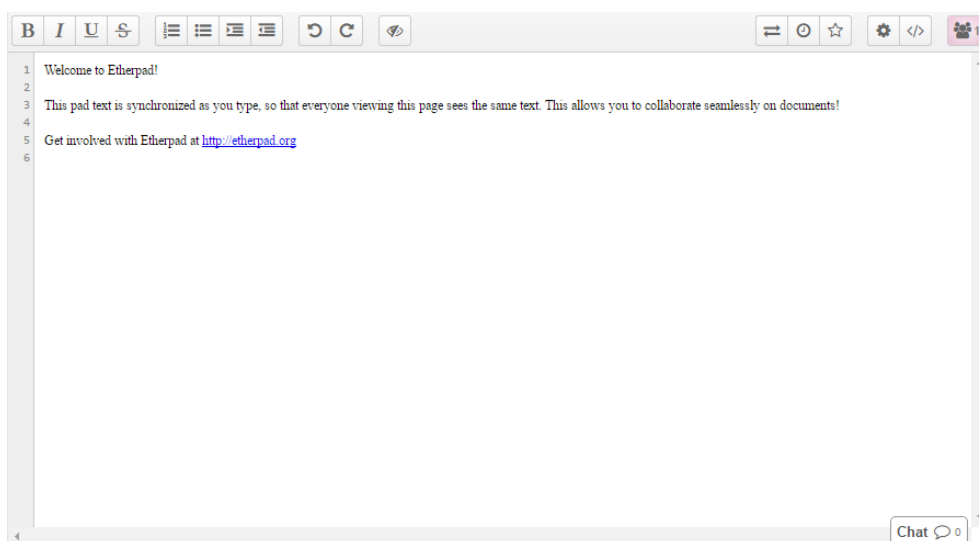


Figura 2.1: Aspetto di un Pad di Etherpad appena aperto

dell'Autore stesso, raffigura tre omini stilizzati affiancati dal numero di Autori. In questo menù è possibile visualizzare l'intera lista di nomi di Autori attualmente collegati, affiancati dal loro colore predefinito. Nella prima riga sono presenti i dati dell'Autore corrente che, da qui, possono essere comodamente modificati, cliccando sul campo di input di testo per modificare il nome o sul quadratino colorato per scegliere un nuovo colore. Se il colore predefinito dell'Autore viene modificato, tutte le modifiche apportate al documento dall'Autore stesso cambieranno colore a loro volta.

Barra degli strumenti

I pulsanti della barra degli strumenti possono essere distinti essenzialmente in due gruppi:

1. le opzioni di formattazione, o funzioni che permettono la modifica del documento e i cui effetti vengono registrati di default ogni volta che li si utilizzano, posizionate sulla sinistra

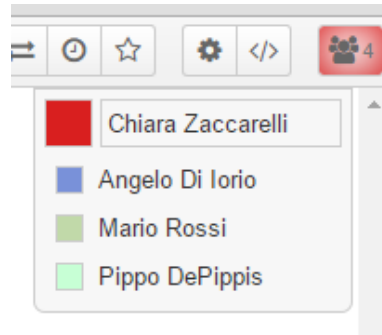


Figura 2.2: Menù degli utenti collegati

2. le impostazioni generali, che invece hanno funzioni più varie, posizionate sulla destra.

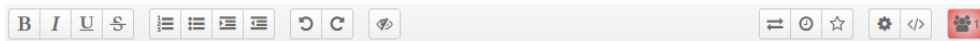


Figura 2.3: Barra degli strumenti di Etherpad Lite

Le opzioni di formattazione del testo di Etherpad Lite sono poche ma essenziali: grassetto, corsivo, testo sottolineato o barrato, oltre alla possibilità di creare liste numerate o non numerate o indentare il testo. A questi sono affiancati anche i bottoni che permettono di annullare o ripetere le azioni appena compiute, nonché il bottone che permette di eliminare in modo permanente l'evidenziatura dei contributi degli Autori fino a quel momento, il cui simbolo sulla barra degli strumenti è un occhio barrato. Quest'ultima opzione, però, non impedisce al testo di essere evidenziato nuovamente quando un Autore decide di fare una modifica successiva al reset dei colori. Un altro modo per eliminare l'evidenziatura del testo si trova sempre nel menù di visualizzazione alla voce "colori che indicano gli Autori". Spuntando tale opzione ogni colore sparirà dal documento: si tratta, tuttavia, di un'operazione diversa da quella precedentemente fatta, in quanto sarà solo una modifica di visualizzazione locale dell'utente, che quindi non verrà trasmessa

anche agli altri Autori. Inoltre, nel caso un Autore decida di modificare una sezione di testo, il Pad continuerà ad apparire senza sottolineature finché non si deciderà di disattivare l'opzione.

Sulla destra, oltre al menù degli Autori attivi, si trovano i bottoni delle impostazioni del Pad. Questi cinque bottoni, sei se si conta anche quello del menù degli Autori, permettono, in ordine, di aprire il menù di importazione ed esportazione, aprire la cronologia, salvare la versione corrente del documento, aprire il menù delle impostazioni generali e infine aprire il menù di condivisione.

Il menù di importazione ed esportazione è identificato da un bottone raffigurante due frecce dirette in direzioni opposte. L'importazione è gestita attraverso un normale form di upload, che permette di caricare dal proprio dispositivo diversi tipi di file di testo semplici, per esempio quelli in formato txt. Altri tipi di file di testo più complessi, come gli odt o i doc/docx di Microsoft, non sono supportati da Etherpad Lite, probabilmente a causa delle opzioni di formattazione che non risultano compatibili. Per quanto riguarda l'esportazione, l'applicazione propone tre formati: un formato "etherpad", in grado di mantenere intatte tutte le caratteristiche del Pad ma che non può essere facilmente utilizzato o modificato al di fuori di un'istanza di Etherpad, oppure i due formati HTML o txt, decisamente più versatili.

Il bottone successivo è quello della cronologia, identificato da un orologio. Facendo click su di esso ci si ritrova in una schermata totalmente nuova: nella parte superiore è presente uno slider che permette, muovendo la barra, di scorrere durante tutta la storia del Pad, il quale verrà visualizzato nella parte sottostante nella versione dettata dallo slider. Da questa schermata non sarà però possibile ripristinare una determinata versione del documento, che potrà invece essere recuperata importando la versione del Pad desiderata se è stata precedentemente esportata o utilizzando la funzione "undo".

Il bottone raffigurante una stella permette di "salvare" la versione corrente del documento. Utilizzando questa funzione, verrà aggiunta sullo slider della cronologia una stella che fungerà da segnalibro e permetterà di raggiungere

agilmente lo stato in cui era il Pad quando si è deciso di salvarlo.

Appena oltre si trova il bottone che apre il menù delle impostazioni, identificato dall'icona di un ingranaggio. Qui è possibile trovare diverse opzioni per personalizzare la visualizzazione del Pad. È possibile fare in modo che la chat e la lista degli utenti risultino sempre visibili o nascondere, come già precedentemente spiegato, l'evidenziazione del testo con i colori degli autori, ma anche nascondere i numeri delle righe oppure allineare il testo a destra, cambiare la lingua dell'interfaccia o il font del documento. Tutte queste impostazioni impattano unicamente la visualizzazione da parte dell'utente stesso.

Ultimo bottone prima di quello degli Autori è quello della condivisione, identificato da una barra dentro due parentesi angolate. Cliccandolo, il menù mostrerà due campi di testo dai quali sarà possibile recuperare rispettivamente il link URL e il codice Embedded per poter condividere il Pad corrente.

Tutto ciò che succede sulla barra degli strumenti viene catturato e registrato nella sezione del database dedicata al monitoring, a partire dai bottoni di formattazione fino a quelli delle impostazioni generali sul Pad e degli elementi che l'Autore può utilizzare negli specifici menù associati.

Chat

Ultima funzionalità dei Pad di Etherpad Lite è la chat, che permette di comunicare con gli altri Autori in tempo reale. Di default la chat può essere aperta grazie ad una linguetta nella parte inferiore dell'interfaccia, ma può anche essere resa sempre visibile selezionando l'opzione relativa nel menù delle opzioni di visualizzazione. Utilizzare la chat è molto intuitivo: per inviare un messaggio basta scrivere nell'apposito campo di input testo e confermare col tasto invio della tastiera.

Anche la chat è un elemento piuttosto interessante da osservare per poter studiare come si comportano gli Autori su uno strumento di questo tipo e per questo è stato scelto di monitorare ogni suo utilizzo. Vengono registrati nel

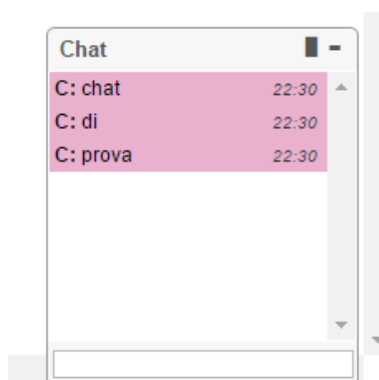


Figura 2.4: Chat di Etherpad

database non solo i messaggi singoli che gli utenti si spediscono a vicenda, ma anche tutte le operazioni che vengono fatte con la chat sull'interfaccia: ovviamente apertura e chiusura, ma anche scrolling dei messaggi ricevuti nel corso del tempo.

2.2.3 UICEM Display

Una volta che è stato raccolto e salvato nel database qualche dato, può entrare in campo il secondo modulo da cui è formato UICEM. UICEM Display, come già spiegato, è una piccola applicazione che recupera i dati dal database MySQL dedicato su cui vengono inizialmente salvati i dati e li organizza in modo leggibile e fruibile, aggregando le informazioni in base a diverse esigenze di chi lo deve utilizzare. Questo strumento può essere utilizzato in modo totalmente indipendente dal precedente, poiché è collegato ad esso solo attraverso il database.

Prima di iniziare ad usare questo strumento, infatti, l'unico requisito necessario è quello di inserire i dati di accesso al database (host, nome utente, password e nome del database) nel form in cima alla pagina. Fatto questo, se l'accesso avviene senza errori, i dati dei Pad verranno caricati e sarà possibile visualizzarli comodamente.

Dati Generali del Database

Una volta che si ha avuto accesso alla visualizzazione dei dati, la prima pagina che appare, corrispondente anche alla prima voce del menù laterale “General”, contiene un breve riassunto delle caratteristiche del database ossia il numero totale di pad, di Autori e di revisioni. Appena sotto è presente un grafico a torta che mostra la distribuzione delle revisioni raggruppate per tipo all'interno dell'intero database. I tipi considerati (da questo grafico e in tutta l'applicazione, se non diversamente specificato) sono quelli relativi ai primi quattro comportamenti descritti prima, dunque cambio di nome, interazioni con la chat, utilizzo delle funzioni di annulla e ripeti e interazione con il menù delle opzioni di visualizzazione.



Figura 2.5: Prima schermata con i dati generali del database

Pad e Autori: dati generali

Dalla schermata principale è possibile selezionare una delle altre voci del menù laterale per iniziare a osservare sempre più nel dettaglio i dati raccolti.

Selezionando la seconda voce è possibile osservare una panoramica dei Pad nel database, mentre selezionando la terza voce è possibile osservare gli Autori. L'interfaccia è la stessa, l'unico cambiamento riguarda, appunto, il soggetto. Se per esempio si osserva la panoramica dei Pad, si può notare

la presenza di due tab. Nel primo tab è presente una tabella in cui in ogni riga è inserito il nome di un Pad seguito dal numero di Autori che vi hanno lavorato, il numero di revisioni totali e il numero di revisioni per ogni tipo. Si fa presente che il numero di revisioni totali può essere diverso dalla somma delle revisioni per tipo visualizzate, in quanto sono conteggiate anche, per esempio, quelle riguardanti lo scrolling. Il secondo tab è invece quello dei grafici: qui è stato inserito un grafico a torta per ogni tipo di revisione riguardante la distribuzione delle revisioni di tali tipi raggruppate per Pad, ognuno organizzato in un sotto tab differente. La panoramica degli Autori funziona in modo assolutamente analogo: nella tabella è stata inserita una riga per ogni Autore contenente nome, numero di Pad a cui ha contribuito, revisioni totali e revisioni per ogni tipo, mentre i grafici rappresentano la distribuzione dei vari tipi di revisione raggruppati per Autore. Si fa inoltre presente che per quanto riguarda gli Autori si è preferito visualizzare l'username scelto dall'Autore stesso attraverso l'operazione di cambio di nome piuttosto che l'identificatore univoco che lo identifica per un fattore di leggibilità. Gli unici Autori visualizzati con il proprio identificatore univoco sono quelli che non hanno mai scelto un username o che hanno un username vuoto.

Pad e Autori: dettaglio

È possibile osservare più nel dettaglio le dinamiche di ogni specifico Pad o Autore presente nelle tabelle semplicemente cliccando sul nome. In questo modo si aprirà una panoramica molto simile a quella appena analizzata, incentrata questa volta sul Pad o sull'Autore scelto. Anche in questo caso sono presenti due Tab, uno coi dati generali in forma tabellare e uno coi grafici. Nel caso del Pad, la tabella inserita nel primo Tab contiene la lista di ogni Autore che ha contribuito a quel determinato Pad affiancato dal numero di sessioni che l'utente ha sostenuto, il numero di revisioni totali e le revisioni per ogni tipo, sempre relative al Pad. Oltre a questo sarà possibile trovare due elementi che non erano presenti nelle pagine di panoramica generale: il primo di questi è il bottone identificato da una piccola freccia orientata a sinistra

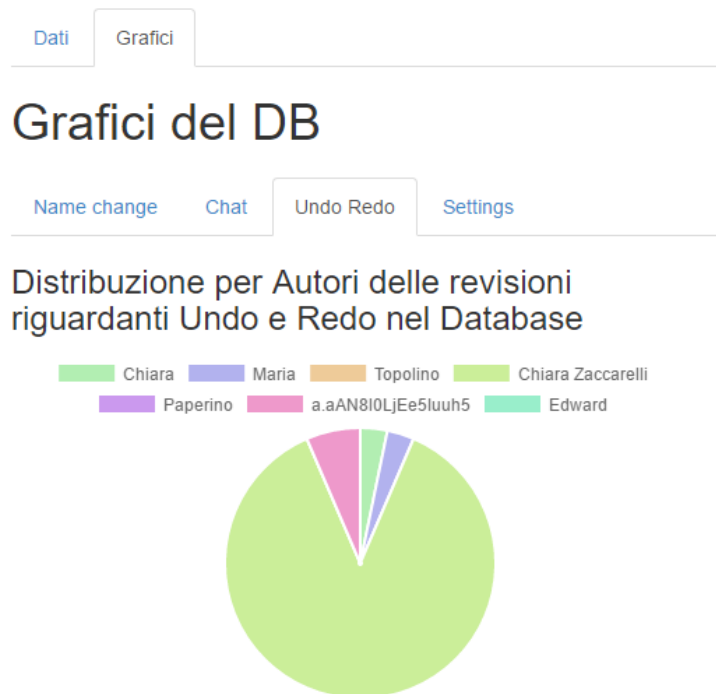
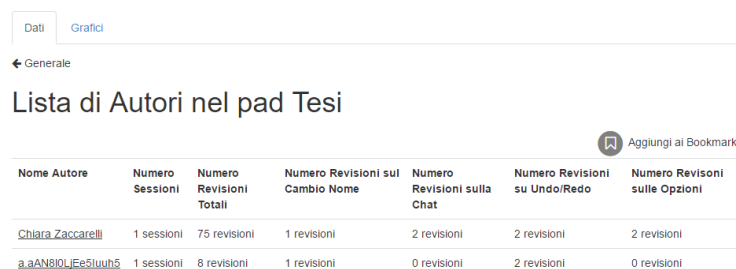


Figura 2.6: Esempio: grafico che indica la distribuzione in tutto il database delle revisioni su Undo e Redo raggruppate per Autore

che permette di tornare al livello superiore (in questo caso la panoramica generale dei Pad) e un bottone che permette di aggiungere o rimuovere il Pad dalla lista dei Bookmark, di cui si parlerà successivamente.

Nel secondo Tab sono presenti cinque grafici a torta: il primo è molto simile a quello mostrato nella pagina “generale” e indica la distribuzione delle revisioni all’interno del Pad raggruppate in base al tipo, mentre gli altri quattro, uno per tipo di revisione, rappresentano la distribuzione delle revisioni di quel tipo all’interno del Pad raggruppate per Autori.

Invece, se si sta controllando un Autore nella tabella è presente una riga per ogni Pad a cui ha contribuito.



Nome Autore	Numero Sessioni	Numero Revisioni Totali	Numero Revisioni sul Cambio Nome	Numero Revisioni sulla Chat	Numero Revisioni su Undo/Redo	Numero Revisioni sulle Opzioni
Chiara Zaccarelli	1 sessioni	75 revisioni	1 revisioni	2 revisioni	2 revisioni	2 revisioni
a.aAN80LJcE5luuh5	1 sessioni	8 revisioni	1 revisioni	0 revisioni	2 revisioni	0 revisioni

Figura 2.7: Esempio: lista autori del Pad “Tesi”

Azioni di un singolo Autore in un singolo Pad

Proseguendo sempre più nel dettaglio, invece, se ad esempio si seleziona l’Autore B a partire dal dettaglio generale dell’Pad A o, viceversa, si seleziona il Pad A dalla tabella dei Pad su cui ha lavorato l’Autore B viene aperto il dettaglio delle azioni dell’Autore B all’interno del Pad A. In questo caso, le righe della tabella rappresentano le varie sessioni. Viene considerata sessione quel lasso di tempo che passa tra il momento in cui l’Autore accede al Pad e quello in cui si disconnette. Per questo motivo le revisioni possono essere smistate anche per sessione, non essendo possibile che una revisione avvenga al di fuori di una sessione poiché significherebbe che l’utente sta agendo sul Pad anche se non è collegato e allo stesso tempo non è possibile che una stessa revisione appartenga a più di una sessione, poiché non possono esistere due sessioni contemporanee dello stesso utente. In sostanza una certa revisione apparterrà ad una e una sola sessione.

In ogni riga, dunque, vicino al numero di sessione è presente, come sempre, il numero di revisioni totali della sessione e il numero di revisioni per ogni tipo. Anche nel caso delle sessioni è presente la possibilità di tornare al livello superiore con la particolarità che, in questo caso, i livelli superiori sono due, essendo due anche i modi in cui si può arrivare a tale dettaglio: sono infatti presenti due bottoni distinti che reindirizzano alla visione generale del Pad o dell’Autore.

Per quanto riguarda il tab dei grafici, in questo caso il grafico è unico,

ossia un grafico a torta contenente la distribuzione delle revisioni dell'autore nel Pad raggruppate per tipo.

Sessioni

Cliccando su una sessione, infine, si giunge finalmente al grado di dettaglio più alto. Anche per il dettaglio della sessione i Tab sono due, ma hanno funzioni diverse rispetto ai precedenti casi. Il Tab principale è quello dei log delle revisioni. Si tratta della lista di tutti i log, che descrivono con precisione quali azioni sono state fatte con annessa data. Sono stati sistemati in modo che risultino raggruppati per tipo e in ordine di data. Ora, oltre ai quattro tipi di log analizzati fino a questo momento si aggiunge anche il quinto, il tipo "Scrolling" che registra un movimento di scroll all'interno del Pad. Il tipo "Scrolling", tra l'altro, trova una sua rappresentazione visiva proprio nel dettaglio sessione, nel secondo Tab. In quest'ultimo Tab, infatti, non ci saranno più grafici a torta rappresentanti i tipi fino ad ora presi in esame, ma un grafico di funzione che scandisce in orizzontale il tempo e in verticale il numero della prima riga visualizzata dall'Autore. In questo modo è possibile osservare con chiarezza quali movimenti di scroll abbia compiuto l'Autore e quando siano stati fatti. È ovviamente possibile anche in questo caso tornare al livello superiore attraverso il bottone apposito.

Bookmark

Come anticipato, l'applicazione presenta anche la possibilità di salvare un determinato Pad o un determinato Autore inserendolo in una lista di Bookmark. In questo modo, soprattutto se il database contiene molti dati, risulta più facile monitorare quelli che sono i Pad o gli Autori più interessanti ai fini della ricerca. Per accedere alle liste occorre cliccare su una delle ultime due voci del menù laterale, rispettivamente "Bookmark Pads" e "Bookmark Authors". Qui è presente una tabella contenente le stesse informazioni che possono essere trovate nella panoramica generale dei Pad o degli Autori, ma unicamente dei Pad che sono stati salvati tra i Bookmark. Anche in

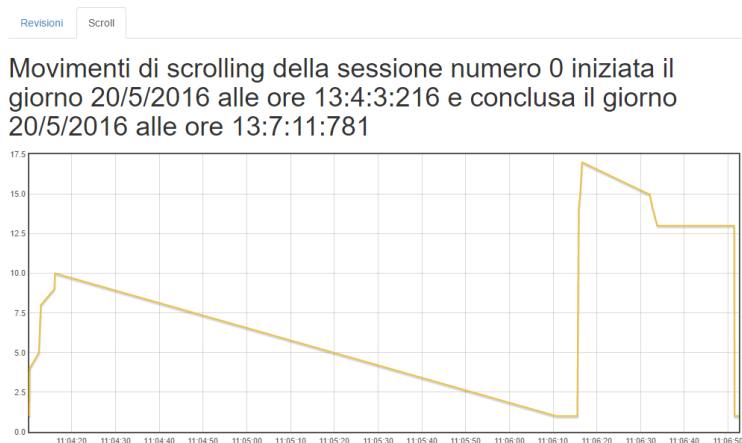


Figura 2.8: Esempio di grafico di scrolling: sull'asse orizzontale il tempo e sull'asse verticale la prima riga visualizzata dall'Autore

questo caso, ovviamente, facendo click sul nome del Pad o dell'Autore si può accedere al dettaglio del Pad o Autore stesso.

La struttura di questa piccola applicazione è stata pensata per agevolare i ricercatori, facendo in modo che risulti facile e intuitivo individuare, raggiungere e studiare i dati raccolti più interessanti.

Lista di Pad nei Bookmarks

Nome Pad	Numero Autori	Numero Revisioni Totali	Numero Revisioni sul Cambio Nome	Numero Revisioni sulla Chat	Numero Revisioni su Undo/Redo	Numero Revisioni sulle Opzioni
<u>Etherpad_Pad</u>	1 utenti	18 revisioni	0 revisioni	0 revisioni	0 revisioni	6 revisioni
<u>test</u>	4 utenti	117 revisioni	7 revisioni	22 revisioni	2 revisioni	11 revisioni

Figura 2.9: Esempio di tabella di Bookmark dei Pad

Capitolo 3

Dettagli implementativi

Come già approfonditamente esposto nel capitolo precedente UICEM è diviso in due moduli molto ben distinti, una versione modificata di Etherpad Lite e uno strumento per visualizzare i dati ottenuti da essa.

Oltre ad avere target, utilizzi e scopi molto diversi, anche la loro implementazione è stata molto diversa: con Etherpad Lite M si è trattato di modificare un programma già esistente prendendo spunto da meccanismi già presenti e modificando passo per passo alcuni file e moduli Node.js utilizzati per scopi simili, mentre per UICEM Display la struttura è stata creata da zero e si è dovuto dare un peso anche alla scelta degli strumenti da utilizzare per implementare tutte le funzionalità, in questo caso ancora javascript con l'ausilio di Node.js e diverse librerie.

3.1 Struttura del Database

Per analizzare la struttura del database è stato necessario innanzitutto migrare il database su un nuovo DBMS. Di default Etherpad Lite utilizza infatti un “dirty.db” di cui però viene sconsigliato l'utilizzo poiché poco pratico in termini di utilizzo: non risulta possibile, infatti, poter interagire con esso se non con l'API costruita ad hoc per Etherpad.

Si è scelto di utilizzare il DBMS MySQL, poiché, senza dubbio, risultava il più pratico a cui interfacciarsi.

Dopo aver spostato il database, andando ad esaminare la sua struttura, la prima cosa che nota un osservatore è certamente la strana organizzazione con cui vengono immagazzinati i dati: ogni singola informazione necessaria al funzionamento dei Pad, che siano dati degli autori, revisioni del Pad o altro, viene inserita nell'unica tabella del DB, chiamata “store”, che, essendo una tabella generale per poter contenere una quantità eterogenea di informazioni, contiene due soli campi dai nomi molto generici, “key” e “value”.

I problemi di questo tipo di approccio appaiono particolarmente visibili quando si prova ad osservare come vengono salvati i vari record: sia la chiave che il valore non risultano certamente immediati da decifrare, poiché contengono molte informazioni tutte insieme.

Quello mostrato di seguito è un esempio di record contenente i dati di un autore:

key	<i>globalAuthor : a.HRBZzTCp6EtL61wI</i>
value	{“colorId” : “c7ff8f”, “name” : “Chiara”, “timestamp” : 1448963252363, “padIDs” : {“Prova” : 1}}

Tabella 3.1: Esempio di record di Etherpad Lite

Si può osservare, dunque, come nel campo “key” venga inserita un’etichetta che indica il tipo di dato, in questo caso “globalAuthor”, seguita dal valore da dare a tale etichetta, dunque “a.HRBZzTCp6EtL61wI”, ossia l’identificativo univoco dell’Autore.

In “value” compaiono, invece, tutti i dati che servono per caratterizzare tale tipo di dato: in questo caso abbiamo a che fare con colorid (il codice del colore assegnato all’Autore), nome, timestamp del primo accesso e lista di Pad a cui ha lavorato.

In tale tabella troviamo, però anche tipi di dati molto diversi, con una chiave strutturata in modo differente e valori caratterizzanti altrettanto ete-

rogenei.

Per fare un altro esempio ecco il record che memorizza una revisione al documento:

key	<i>pad : 8bDXg7Rl8N : revs : 14</i>
value	{ <i>“changeset” : “Z : e > 1 = d * 0 + 1\$M”</i> , <i>“meta” :</i> <i>{“author” : “a.YP0tCfsiuroKOj02”</i> , <i>“timestamp” : 1465853051463}}</i> }

Tabella 3.2: Esempio di record di Etherpad Lite

Anche in questo caso nel campo “key” si trova l’etichetta che identifica il tipo di informazione immagazzinata, in questo caso “pad” e “rev” che indicano che si tratta di una revisione su un certo Pad, con i valori a questa assegnati, il nome del Pad “8bDXg7Rl8N” e il numero di revisione 14. Il campo “value” contiene invece dati riguardanti le modifiche fatte al testo, l’Autore che ha apportato tali modifiche e un timestamp.

Come si può notare dal confronto di questi due record i tipi di informazioni immagazzinati nella stessa tabella possono essere estremamente diversi tra loro e ciò rende l’estrazione dei dati piuttosto complessa e molto poco intuitiva, rendendo necessario, lato applicativo, il dover predisporre alcune funzioni che dividendo i campi opportunamente, riescano a recuperare le variabili giuste; tale inconveniente può essere ovviato con un’organizzazione più strutturata con svariate tabelle e diversi campi per diversi elementi da salvare: in questo modo l’estrazione di una singola informazione potrebbe essere raggiungibile con una semplice query.

Chiaramente ripensare al modo in cui Etherpad Lite salva i propri dati in modo nativo esula dal problema principale, ma osservare e analizzare i problemi di tale struttura mi è stato utile per capire che, almeno per le informazioni aggiuntive sul monitoring delle azioni dell’Autore sull’interfaccia di Etherpad Lite, sarebbe stato necessario un approccio diverso. Proprio per questo motivo, è stata creata una nuova tabella chiamata, appunto, “monitor”.

La tabella risulta strutturata in modo più leggibile e accessibile rispetto alla “store” predefinita e divide ogni record in cinque campi. Il primo campo, “id”, è riservato ad un identificatore numerico progressivo. È stato pensato di utilizzare un numero progressivo di tale genere, in quanto semplifica di molto i calcoli circa la determinazione delle varie sessioni di ogni utente, rendendo possibile un ordinamento dei dati estratti per data dal più vecchio al più recente. Questo aspetto verrà approfonditamente spiegato in seguito. Il nome del Pad viene invece inserito in un campo specifico, chiamato appunto “pad”. Similmente nel campo “Autore” viene salvato il nome identificativo dell’Autore (che è diverso dal nome scelto dall’Autore, poiché quello può essere modificato in qualsiasi momento dall’Autore stesso e potrebbe dunque non essere univoco). Ci sono infine il campo “Timestamp”, in cui viene inserito il momento in cui viene compiuta l’azione, e il campo “Azione”. Nel campo azione viene inserito il nome dell’azione insieme ad, eventualmente, alcuni dati aggiuntivi. In generale il campo “azione” risulta molto descrittivo nei suoi contenuti. Per esempio quando un Autore fa scroll sul documento e si ritrova ad avere come prima riga la riga 15 nel campo azione viene registrato un record con campo azione uguale a “pad scrolling: line 15” oppure se un utente cambia nome da Pippo a Paperino nel record il campo azione viene registrato come “name change from Pippo to Paperino” e così via.

I nuovi record avranno dunque un aspetto di questo genere:

chiave	147
pad	<i>EtherpadPad</i>
azione	<i>undo</i>
utente	<i>a.YP0tCfsiuroKOj02</i>
timestamp	1463742390671

Tabella 3.3: Esempio di log che registra un evento di tipo “undo” sul Pad “Etherpad Pad” dell’Autore con identificativo univoco “a.YP0tCfsiuroKOj02”

3.2 Etherpad Lite M

Avendo, ora, chiara la struttura in cui i dati vengono salvati ci si può concentrare sul come vengono salvati.

Etherpad Lite, come si è visto, utilizza Node.js. Quest'ultimo è strutturato in modo tale da utilizzare la maggior parte dei propri moduli a lato server, mentre risulta decisamente più complicato creare una connessione tra il Client e un modulo come può essere per esempio quello di MySQL.

Per questo motivo, per poter salvare le nuove informazioni, si è dovuto creare un flusso delle stesse che prendesse spunto dai flussi di informazioni già esistenti, utilizzando funzioni costruite sul modello di altre già presenti, ma che si differenzino abbastanza da poter salvare in modo indipendente e corretto tutti i nuovi dati.

Per farlo, il primo passo è stato osservare come vengono trasportati altri tipi di dati, per esempio i messaggi della chat, dall'interfaccia al database. Quando un Autore decide di scrivere un messaggio nella chat, viene inviato un oggetto al CollabClient contenente il tipo di messaggio (in questo caso "CHAT.MESSAGE") e il testo del messaggio. Allo stesso modo, nel caso dei messaggi di monitor, si è pensato di utilizzare il CollabClient allo stesso modo ma differenziando questo tipo di messaggio da quello della chat utilizzando un nuovo tipo, battezzato per semplicità "MONITOR.MESSAGE".

Il CollabClient è a sua volta un oggetto contenuto all'interno dell'oggetto "Pad", grande contenitore che include tutte le funzioni necessarie alla gestione del Pad corrente e tutti i suoi dati. Tra le altre funzioni c'è anche quella che permette di gestire i vari tipi di messaggio inviati al CollabClient richiamando la funzione "HandleMessage".

La funzione HandleMessage esamina il tipo di messaggio e lo invia alla funzione apposita. Nel caso dei messaggi di Chat questa funzione è denominata "handleChatMessage", mentre nel caso dei messaggi di monitor alla nuova funzione è stato dato il nome di "handleMonitorMessage". Queste funzioni di handling permettono di recuperare i dati mancanti (nome dell'Autore, del Pad e creare un Timestamp, per esempio) e di inviarli a una funzione ("ap-

pendChatMessage” per i messaggi di Chat e dunque “appendMonitor” per i messaggi di Monitor) che si occupi di creare l’oggetto contenente l’insieme completo dei dati da salvare sul database e inviarli al server, o meglio, al modulo ueberdb utilizzato dal Server per interfacciarsi con il Database.

Generalmente per tale attività tutti gli altri tipi di messaggi utilizzano la funzione set, che però non può essere utilizzata per salvare i messaggi di monitoring, poiché li salverebbe nella tabella “store” nel formato precedentemente illustrato. Per questo motivo è stata creata una nuova funzione “setmonitor”, che, al contrario, indirizza il contenuto al file javascript che si occupa di comunicare con il tipo di Database in uso (nel nostro caso MySQL) e di fare la query che aggiorni il Database e salvi definitivamente l’azione che l’Autore ha compiuto sull’interfaccia nel database.

Dunque come si è visto, è stato utilizzato il flusso di informazioni già presente per controllare altri tipi di messaggio come ispirazione per salvare anche le nuove informazioni.

Nella tabella Monitor, in ogni caso, vengono salvati molti interventi diversi sull’interfaccia. Ogni operazione sui bottoni viene puntualmente monitorata, anche se solo alcuni di questi vengono poi analizzati nello strumento di visualizzazione. Allo stesso modo, vengono monitorati anche tutti i gli interventi fatti sulla chat (quando viene aperta, chiusa o viene fatto uno scroll sul contenuto) e lo scrolling sul documento. Viene infine salvato un record ogni volta che un Autore si connette o disconnette al Pad. Questi due ultimi elementi riguardano informazioni essenziali per poter determinare e calcolare le varie sessioni.

3.3 UICEM Display

Per quanto riguarda UICEM Display, ci troviamo dunque davanti ad un’implementazione del tutto nuova. Il primo passo è stato quello di decidere con quali linguaggi sviluppare questo strumento. Alla fine ho scelto

di utilizzare javascript e il suo ambiente Node.js per avere continuità tra il primo e il secondo modulo.

Essendo comunque una piccola applicazione browser è stato necessario implementare sia un piccolo server, che si occupi di gestire la comunicazione con il Database, che un Client che svolga il compito effettivo di visualizzare i dati sullo schermo in modo organico.

3.3.1 Recupero e Organizzazione dei Dati

Il recupero dei dati avviene, dunque, esclusivamente attraverso il server. Dopo che l'utente ha inserito i dati per accedere al database nel form presente sull'interfaccia del Display, questi possono essere inviati al server che li utilizza per stabilire la connessione.

Dopo aver eseguito l'anzidetta operazione, con una query vengono estratti tutti i dati presenti nella tabella "monitor" (ordinandoli per chiave primaria ossia, siccome la chiave è un intero con valore "AUTO INCREMENT", dalla riga più vecchia alla più nuova) e, dopo aver inserito record in un array, li invia al client.

L'estrazione dei dati dal database avviene, dunque, solamente all'inizio e per aggiornare gli stessi dopo che il client li ha già ricevuti è necessario ricaricare e inviare una seconda volta i dati di accesso del database al server.

Siccome il client riceve tutti i record così come sono salvati sul database è sua responsabilità quella di trovare un modo efficiente per trattare i dati, in modo che gli stessi non debbano essere calcolati ed esaminati ogni volta che le informazioni da visualizzare nella pagina devono essere sostituite.

Proprio per questo motivo, appena i dati vengono portati sul client vengono creati due array in cui le informazioni che devono essere visualizzate vengono immediatamente smistate, calcolate e nidificate.

Nel primo array (chiamato "epad") vengono inseriti i Pad: ogni oggetto dell'array rappresenta un Pad e contiene il nome del Pad, numero di Autori che hanno lavorato sul Pad, numero di Revisioni totali e numero di Revisioni per ogni tipo. Inoltre è presente anche un Array in cui ogni oggetto rap-

presenta uno degli Autori che ha lavorato al Pad. Ogni elemento di questo array conterrà quindi il nome dell'Autore, sia quello identificativo che quello visualizzato, il numero di revisioni totali e per tipo, il numero di sessioni su quel Pad, un campo che rappresenta l'indice dell'autore stesso nel secondo grande array e due ulteriori array di oggetti i quali rappresentano la lista delle sessioni e quella delle revisioni. La prima contiene i dati riguardanti l'inizio e la fine della sessione e il numero di revisioni totali e per tipo. La seconda contiene un elemento per ogni tipo di revisione (name change, scroll, chat, undo/redo e settings) più uno ulteriore nel quale sono stati inseriti i record riguardanti le sessioni. Ogni elemento contiene la lista di tutte le revisioni del tipo che rappresenta. L'elemento delle sessioni contiene inoltre alcune variabili che tornano utili nel momento in cui si devono calcolare le statistiche delle sessioni stesse.

Il secondo array in cui vengono smistate le informazioni (chiamato "eauth") è quello degli Autori, che risulta un po' meno articolato rispetto a quello dei Pad. Ogni elemento dell'Array contiene i nomi dell'Autore (sia quello identificativo che quello visualizzato), il numero di revisioni totali e per tipo, il numero di Pad su cui ha lavorato e un ulteriore array riguardante il dettaglio di tali Pad. Inserire qui tutte le informazioni sulla relazione tra un Pad e un Autore risulterebbe, però, ridondante, poiché queste informazioni sono già presenti nell'Array dei Pad "epad". Per questo motivo ogni voce contiene, sì, il nome del Pad, ma, invece delle varie informazioni sulle sessioni o sul numero di revisioni, sono presenti due indici che permettono di recuperare l'oggetto contenente i dati richiesti all'interno dell'array generale dei Pad.

La motivazione di base di questo sistema, che magari può apparire un po' complesso, è in realtà molto semplice: lo stesso risulta molto efficiente. È vero che i dati non vengono mai aggiornati, ma l'aver i risultati delle operazioni più frequenti già calcolate e a portata di mano (basta conoscere gli indici in cui sono posizionate le informazioni che interessano per avere accesso immediato a tutti i dati necessari) e non dover richiamare il database ogni volta che si ha bisogno di una nuova informazione rende l'applicazione

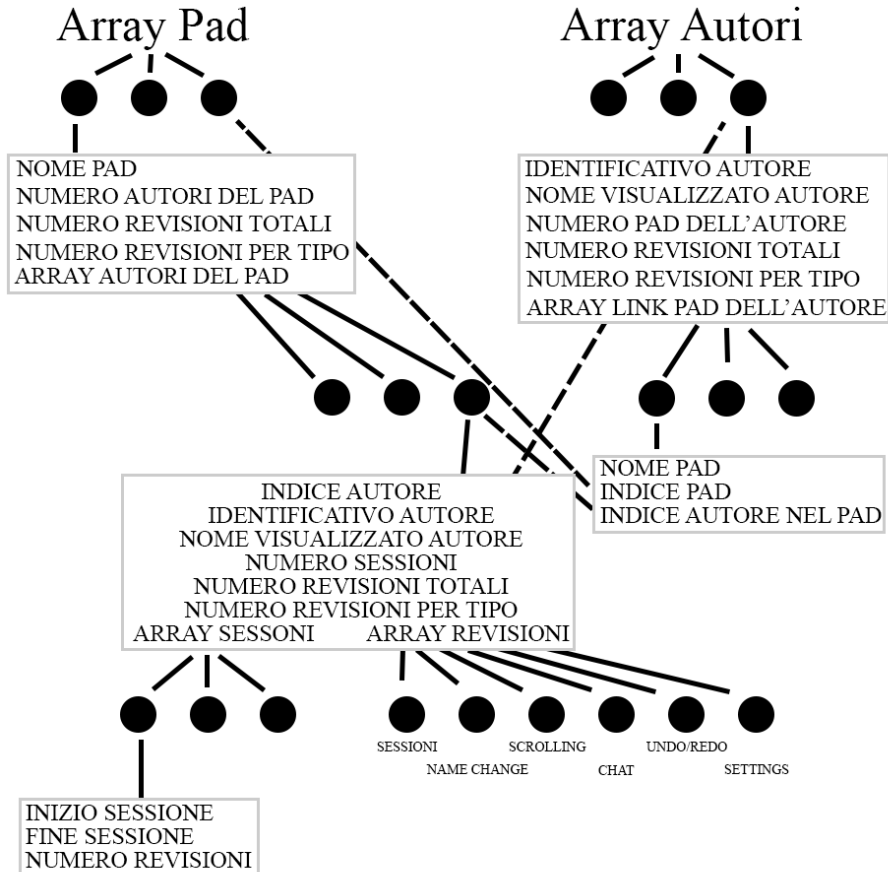


Figura 3.1: Schema che descrive la struttura degli array contenenti i dati

decisamente più veloce e semplice da gestire.

3.3.2 Sessioni

Come già spiegato nel capitolo precedente, nel database, oltre alle informazioni riguardanti bottoni dell'interfaccia e scrolling, vengono salvate anche tutte le volte in cui un Autore si connette e si disconnette da un Pad.

Si è anche già spiegato che le sessioni sono definite come quei periodi di tempo che si collocano tra un log di connessione al Pad e uno di disconnes-

sione. Questo è sempre verificato, poiché non può esistere più di una sola sessione contemporanea di uno stesso Autore su uno stesso Pad: se un Autore tenta di aprire lo stesso Pad due volte, infatti, invece di consentire due sessioni contemporanee, Etherpad disconnette automaticamente la prima a favore della seconda aperta.

Per calcolare una sessione si è deciso di utilizzare un metodo molto semplice: siccome i record vengono estratti in ordine temporale (utilizzando l'indice progressivo) sicuramente il primo log di disconnessione dopo uno di connessione sarà quello che chiude la sessione.

Se non ci sono stati problemi di registrazione dei log di connessione e di quelli di disconnessione (può capitare nel caso il server di Etherpad venga fatto terminare prematuramente prima che l'Autore si sia disconnesso) le sessioni saranno sempre registrate in modo corretto.

3.3.3 Bookmarks

Si è già parlato di come i bookmark si usano, dunque in questo breve paragrafo mi soffermerò sul loro funzionamento.

I bookmark, essendo elementi che devono risultare recuperabili ogni volta che il Display viene riavviato, hanno la necessità di essere salvati a loro volta su un database. Per questo è stata costruita una nuova tabella, chiamata semplicemente “bookmarks”, fatta apposta per contenere i dati dei bookmark.

Questa tabella è formata da quattro campi:

1. “ID”
2. “Type”
3. “User”
4. “Value”

In primo luogo una chiave primaria “ID”, rappresentata semplicemente da un numero progressivo, con l’unico scopo di essere utilizzato come identificativo del record.

Il campo “Type” invece, indica il tipo di bookmark rappresentato dal record e ha valore 0 se ci si riferisce ad un Pad, mentre ha valore 1 se il soggetto è un Autore.

Il campo “User” contiene il nome dell’utente del Database che sta utilizzando il Display (recuperato in base ai dati che l’utente stesso ha utilizzato per entrare in contatto con il database), in modo che ogni utente del Database possa avere i propri bookmark.

Il campo “Value”, infine, contiene il nome del Pad o dell’Autore da salvare.

I record risultano, dunque, strutturati in modo simile a come illustrato nella tabella 3.4.

ID	133
user	<i>root</i>
type	1
value	<i>a.P8N4psc85AmRNpvo</i>

Tabella 3.4: Esempio di log di Bookmark in cui l’utente root ha salvato l’Autore (type=1) con identificativo univoco a.P8N4psc85AmRNpvo

Una volta estratta la lista dei bookmark dell’utente Database corrente, questa viene caricata su un array chiamato “bookmark” appena si effettua il collegamento con il database (sempre e solo dopo aver popolato i due array epad ed eauth), proprio come avviene nel caso delle normali revisioni.

Questo nuovo array deve essere aggiornato ogni volta che l’utente decide di aggiungere o togliere un elemento. Inoltre questo array viene sempre tenuto sotto controllo anche quando si accede al dettaglio di uno specifico Pad o Autore: infatti, se il Pad o l’Autore è già presente nell’array, viene caricato

il bottone per rimuovere il soggetto dai bookmark, mentre se non è presente viene visualizzato il bottone che permette, al contrario, di aggiungerlo.

3.3.4 Grafici

Trasportare i dati già immagazzinati in un array sotto forma testuale, che si tratti di inserirli in una tabella o meno, non risulta particolarmente complicato.

Per creare dei grafici, invece, se si vuole fare un buon lavoro è necessario affidarsi a librerie esterne.

UICEM Display contiene essenzialmente due tipi di grafico, quello a torta dei dati aggregati e quello lineare riguardante le operazioni di scrolling nella sessione.

I grafici a torta sono stati costruiti utilizzando una libreria Javascript chiamata “chart.js”¹, mentre quelli per lo scrolling durante le varie sessioni sono stati creati con un’altra libreria, “flot”².

Si sono rivelate entrambe strumenti molto immediati da utilizzare, poiché i dati che devono essere rappresentati possono essere caricati attraverso un comodissimo array di oggetti Javascript. Inoltre i grafici risultanti appaiono a loro volta molto intuitivi grazie ad un’organizzazione della legenda molto chiara.

¹<http://www.chartjs.org/>

²<http://www.flotcharts.org/>

Capitolo 4

Valutazione

4.1 Test

Allo scopo di analizzare il modo in cui UICEM si comporta, ho deciso di organizzare alcune sessioni di test.

Si è trattato di due sessioni dalle caratteristiche molto diverse.

La prima si è svolta in mezz'ora con un gruppo composto da 4 persone a cui era stato assegnato il compito di descrivere ciò che accadeva durante un programma televisivo precedentemente scelto, programma che tutte e quattro le persone stavano guardando nello stesso momento.

Nella seconda sessione, invece, due Autori avevano il compito di correggere e completare la scrittura di un documento incompleto in un quarto d'ora.

A partire dai dati prodotti da tali simulazioni si è tentato di identificare sia i punti di forza che quelli di miglioramento dello strumento.

Nell'immagine 4.1 è possibile osservare gli aspetti più importanti che differenziano i Pad utilizzati per i due test (“test1” rappresenta il primo test e “test2” il secondo).

Nell'immagine 4.2, invece, è possibile osservare come lo strumento abbia identificato i quattro autori che hanno partecipato al primo test e registrato ogni loro azione diretta a comunicare con l'interfaccia.


Dati Grafici

Lista di pad nel Database

Nome Pad	Numero Autori	Numero Revisioni Totali	Numero Revisioni sul Cambio Nome	Numero Revisioni sulla Chat	Numero Revisioni su Undo/Redo	Numero Revisioni sulle Opzioni
test1	4 utenti	319 revisioni	3 revisioni	13 revisioni	1 revisioni	5 revisioni
test2	2 utenti	202 revisioni	3 revisioni	54 revisioni	4 revisioni	3 revisioni

Figura 4.1: Quadro generale dei Pad su cui si sono svolti i test

Lista di Autori nel pad test1

 Rimuovi dai Bookmark

Nome Autore	Numero Sessioni	Numero Revisioni Totali	Numero Revisioni sul Cambio Nome	Numero Revisioni sulla Chat	Numero Revisioni su Undo/Redo	Numero Revisioni sulle Opzioni
C	1 sessioni	66 revisioni	0 revisioni	5 revisioni	1 revisioni	5 revisioni
A	1 sessioni	80 revisioni	1 revisioni	4 revisioni	0 revisioni	0 revisioni
D	1 sessioni	92 revisioni	1 revisioni	2 revisioni	0 revisioni	0 revisioni
B	1 sessioni	81 revisioni	1 revisioni	2 revisioni	0 revisioni	0 revisioni

Figura 4.2: Quadro generale del Pad “test1”

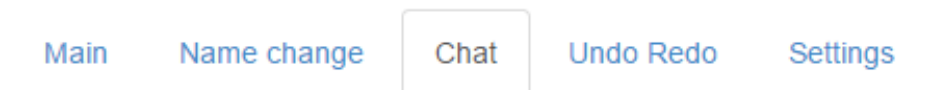
Da una schermata come questa si può vedere come le revisioni riguardanti Annulla e Ripristina siano state poche, un solo Autore ha usato tale funzionalità. Allo stesso modo, anche le Opzioni di Visualizzazione non sono state utilizzate molto. Il cambio di nome, invece, è stato utilizzato da quasi tutti gli Autori e la chat è stata aperta e/o chiusa almeno un paio di volte da tutti.

Come risulta nell’immagine 4.3, inoltre, i dati visualizzati dai grafici sono esattamente quelli riportati nella tabella relativa.

Nell’immagine 4.4 è possibile osservare inoltre, i dettagli del Pad su cui si è svolto il secondo test.

Grazie ai log situati nel dettaglio di ogni sessione si può inoltre controllare

Grafici del pad test1



Distribuzione per Autore delle revisioni riguardanti la chat nel Pad test1

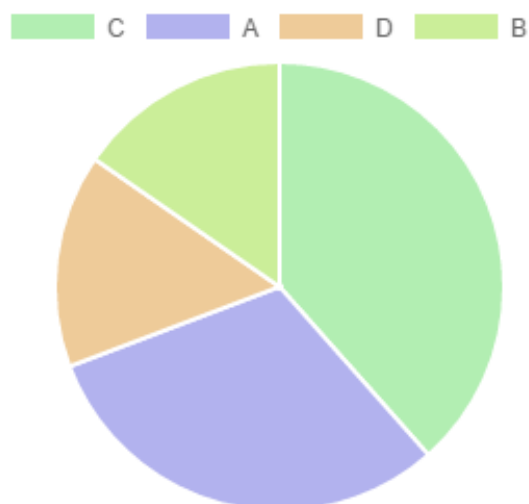


Figura 4.3: Grafico relativo alle revisioni della chat del Pad “test1”


quando queste operazioni siano state compiute e quali effetti abbiano portato nello specifico.

Si può, per esempio leggere che tutti i cambi di nome sono avvenuti all’inizio della sessione oppure che l’Autore B, nonostante mostri due sole revisioni per quanto riguarda l’utilizzo della chat, ha tenuto la chat aperta per tredici minuti.

Altri dettagli interessanti possono essere notati nell’osservare il grafico degli scroll, sempre nel dettaglio della sessione.

Dal confronto tra i quattro grafici nella figura 4.5 si può notare come

Lista di Autori nel pad test2

 Rimuovi dai Bookmark

Nome Autore	Numero Sessioni	Numero Revisioni Totali	Numero Revisioni sul Cambio Nome	Numero Revisioni sulla Chat	Numero Revisioni su Undo/Redo	Numero Revisioni sulle Opzioni
<u>Y</u>	1 sessioni	91 revisioni	2 revisioni	27 revisioni	3 revisioni	2 revisioni
<u>X</u>	1 sessioni	111 revisioni	1 revisioni	27 revisioni	1 revisioni	1 revisioni

Figura 4.4: Quadro generale del Pad “test2”

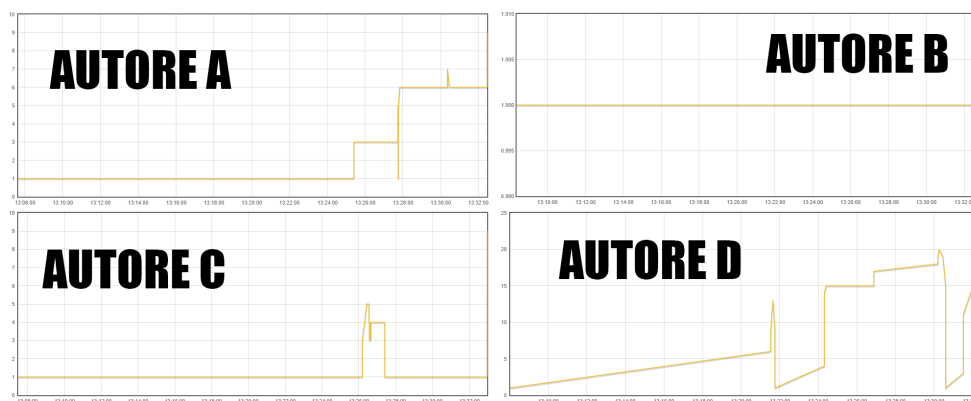


Figura 4.5: Confronto tra le operazioni di scrolling dei vari Autori del Pad “TVTest”

ogni Autore si sia mosso all’interno del pad in modo completamente diverso: l’Autore A, per esempio, si è mosso quasi sempre verso il basso, mentre gli autori C e D sono tornati all’inizio a rileggere e, addirittura, l’autore B è rimasto sempre costante nello stesso punto.

Il test, tuttavia, è stato fatto su piccola scala e dunque non è stato in grado di dimostrare tutte le potenzialità dello strumento. Ha fornito, però, abbastanza elementi per determinare alcune migliorie che potrebbero risultare utili.

4.1.1 Criticità

Purtroppo non si è in grado di determinare quali siano le problematiche specifiche dell'UICEM Display nel caso i dati da lavorare siano una grande quantità. Certo è che una elevata numerosità di dati comporta anche molto lavoro al fine di smistare i vari log e, conseguentemente, una lunga elaborazione.

Allo stesso tempo, avere una grande quantità di dati comporta anche la necessità di avere tabelle molto lunghe dalle quali risulta complicato estrarre le specifiche informazioni.

UICEM Display potrebbe, dunque, trovare difficoltà nel dover gestire ed esporre efficacemente una tale abbondanza di informazioni.

4.2 Miglioramenti

Dopo aver analizzato i risultati di questi test e le principali criticità, l'obiettivo in questa sezione è capire quali elementi possano essere migliorati per rendere meno evidenti tali problematiche.

4.2.1 Filtri di Ricerca

Innanzitutto per rendere i dati ancora più fruibili e favorire il rintracciamento di determinati dati, specialmente nel caso in cui le informazioni siano in grandi quantità, potrebbe risultare utile avere qualche tipo di filtro o di ricerca per selezionare solo tipi di elementi che soddisfano determinate caratteristiche. Grazie ad una ricerca potrebbero essere immediatamente evidenti, ad esempio, in quali Pad è stata utilizzata maggiormente la chat, quali Autori hanno cambiato più spesso il proprio username o in quali sessioni gli scroll sono stati più frequenti.

4.2.2 Integrazione tra meccanismi di editing e interfaccia

L'obiettivo della ricerca è quello di capire come si muove un utente quando utilizza un Editor Collaborativo. Questo significa che l'interazione con l'interfaccia è solo una parte del lavoro che deve essere analizzato. Per rendere l'analisi più completa, si potrebbe pensare di integrare ad UICEM anche l'analisi dei meccanismi di editing, così da poter controllare quando e in che modo un certo Pad sia stato modificato e se tale modifica abbia causato degli effetti sul modo in cui un Autore si sia mosso sull'interfaccia e viceversa.

Tale integrazione, però, causerebbe anche un grosso supplemento della quantità di informazioni e lo strumento potrebbe avere difficoltà a gestirle tutte correttamente.

4.2.3 Confronto tra due o più elementi dello stesso tipo

Un'altra funzionalità che potrebbe risultare particolarmente stimolante è quella che permette di scegliere due o più elementi dello stesso tipo, per esempio due Pad, due Autori, due Sessioni e così via e poterle analizzare fianco a fianco, sia per quanto riguarda i dati che in forma di grafico, magari rendendo disponibili grafici che mettano bene in evidenza quali sono le differenze che corrono tra un elemento e l'altro.

Penso che questa funzionalità avrebbe un utilizzo interessante, per esempio, nel determinare durante una particolare sessione a cui due Autori hanno partecipato se ad un'azione di uno ne sia conseguita una dell'altro e viceversa.

4.2.4 Estensione ad altri Editor

Come già illustrato, UICEM Display recupera le informazioni che deve analizzare da un database al quale si deve accedere eseguendo un login ogni volta che lo si vuole utilizzare e per questo non è legato ad uno specifico database o ad una specifica istanza di Etherpad Lite M. Proprio per questo

motivo, tale strumento potrebbe risultare riutilizzabile nel caso si volesse fare un'analisi simile su un Editor diverso.

Ovviamente in questo caso sarebbe necessaria un'attenta analisi dell'interfaccia di tale Editor, poiché in alcuni casi potrebbero mancare funzionalità che sono invece presenti in Etherpad Lite, come ad esempio la possibilità di cambiare nome o di utilizzare una chat.

4.3 Usabilità e Mantenimento del Codice

Dopo aver analizzato grazie al test quelli che sono i punti di forza e i punti di miglioramento sul software, mi soffermerò su quello che è a mio parere un altro dei punti a favore di questo progetto, soprattutto se si pensa alle eventuali implementazioni e modifiche future, ossia come il codice è stato scritto e come viene mantenuto strutturato.

Innanzitutto, per quanto riguarda Etherpad Lite M, ho tentato di seguire, passo passo tutte le funzioni scritte dagli sviluppatori della versione standard di Etherpad Lite o di crearne di nuove che utilizzassero la stessa struttura. In questo modo il codice risulta certamente ben amalgamato al resto.

Per quanto riguarda UICEM Display, strutturare il codice partendo da zero è risultato un po' meno immediato. Per rendere il lavoro di sviluppo più agevole ho deciso di utilizzare il task runner Gulp¹.

Gulp viene utilizzato principalmente per due finalità:

1. rendere il codice client side modulare;
2. gestire e riavviare il server ogni volta che viene fatta una modifica al codice.

Come si evince dalla descrizione svolta nel capitolo 4, la maggior parte del lavoro dello strumento di visualizzazione avviene client side e per questo, per poter avere un codice comprensibile e ben strutturato, è necessario poterlo

¹<http://gulpjs.com/>

dividere in diversi file che però, successivamente, devono essere dichiarati singolarmente uno ad uno. Grazie a Gulp, invece, tutti i file in cui è stato scritto il client vengono automaticamente uniti in un unico file “all.js” che viene aggiornato ogni volta che viene creato un nuovo file di codice Javascript per il client e ogni volta che uno dei file già esistenti viene modificato. Tale file viene costruito anche grazie all’utilizzo di Browserify² che, analizzando i singoli file di partenza, permette di ordinare gli scripts in modo tale da evitare problemi con le dipendenze che, unendo semplicemente parti di codice senza stabilire un ordine, rischierebbero di emergere ogni volta. In questo modo il codice client side può avere un’agevole struttura modulare e, allo stesso tempo, non ci si deve preoccupare troppo di dichiarare tutti i file creati, poiché l’unico file che necessita di essere dichiarato è “all.js”. Inoltre avere un unico file rende più agevole il caricamento della pagina stessa.

Il secondo compito per cui Gulp è risultato piuttosto utile è stata la gestione del server. Grazie ad una funzionalità chiamata LiveReload, ogni volta che uno dei file sorgenti viene modificato, il server viene riavviato. In questo modo l’implementazione di nuove funzionalità e la correzione dei bug diventa ancora più veloce, poiché non è più necessario riavviare il server manualmente ogni volta.

Dunque task runner come Gulp mi ha fornito un grande aiuto durante lo sviluppo e credo che il suo utilizzo fornisca un valore aggiunto all’applicazione, il cui codice risulta facile da mantenere e comprendere.

²<http://browserify.org/>

Conclusioni

Lo scopo di questa dissertazione era quello di dimostrare che può essere possibile trovare un modo per ottenere numerosi dati dall'interfaccia di un servizio RTCE, in questo caso Etherpad Lite, allo scopo di determinare come un utente agisce su tale strumento.

Per farlo è stato sviluppato uno strumento (Etherpad Lite M) modificando l'Editor stesso in modo che, proprio come fa con le modifiche ai documenti, possa catturare e salvare sul database tutte le informazioni riguardanti le azioni dell'utente sull'interfaccia. Queste informazioni, inoltre, vengono salvate in modo strutturato cosicchè non risulti arduo il loro recupero (a differenza dei log dello stesso Etherpad, che inserendo troppe informazioni nei due soli campi molto generici dei record necessita di essere decifrato dall'applicazione che ne deve fare uso). Questo aspetto è molto importante poiché si è pensato di costruire un secondo strumento (UICEM Display), che a sua volta deve interfacciarsi al database. Questo tool deve recuperare i dati, elaborarli e mostrarli all'utente finale in modo che possa utilizzarli e studiarli in modo agevole.

In generale, i tipi di dato che vengono presi in considerazione (cambio di nome, utilizzo della chat, annulla/ripeti, impostazioni di visualizzazione e scrolling) sono solo alcuni dei tipi di dato che, come si è visto, possono essere recuperati da Etherpad Lite M: in questa prima implementazione si è scelto di concentrarsi su questi, in quanto sono stati ritenuti i più rilevanti per lo studio a cui questo progetto tenta di dare un contributo. L'idea è, eventualmente, quella di integrare anche tutti gli altri tipi di interazione con

l'interfaccia, in modo tale che l'applicazione possa dare un quadro davvero completo riguardo le azioni che un Autore compie sull'interfaccia.

A questa, vanno ad aggiungersi le possibili migliorie che a mio parere potrebbero risultare utili da implementare in una nuova versione di questo software, ossia la possibilità di fare ricerche in modo da filtrare le informazioni, l'integrazione dei dati riguardanti l'interfaccia con quelli relativi ai meccanismi di editing, la possibilità di poter confrontare due o più elementi affiancando i dati, calcolando le differenze e esprimendole attraverso grafici ad hoc e un eventuale estensione ad altri editor qualora fosse necessario.

Bibliografia

- [1] Mehdi Ahmed-Nacer, Claudia-Lavinia Ignat, G'erald Oster, Hyun-Gul Roh, and Pascal Urso *Evaluating CRDTs for real-time document editing*. In Proceedings of the 2011 ACM Symposium on Document Engineering, pages 103-112, 2011.
- [2] Jeremy Birnholtz and Steven Ibara. *Tracking changes in collaborative writing: Edits, visibility and group maintenance*. In Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work, CSCW '12, pages 809-818, 2012.
- [3] Jeremy P. Birnholtz, Stephanie B. Steinhardt, and Antonella Pavese. *Write here, write now!: an experimental study of group maintenance in collaborative writing*. In 2013 ACM SIGCHI Conference on Human Factors in Computing Systems, CHI '13, pages 961-970, 2013.
- [4] Cornelia Brodahl and Nils Kristian Hansen. *Education students' use of collaborative writing tools in collectively reflective essay papers*. Journal of Information Technology Education: Research, 13:91-120, 2014.
- [5] Sam Kai Wah Chu and David M. Kennedy. *Using online collaborative tools for groups to co-construct knowledge*. Online Information Review, 35(4):581-597, 2011.
- [6] Paul Dourish and Victoria Bellotti. *Awareness and coordination in shared workspaces*. In Proceedings of the ACM 1992 Conference on

- Computer Supported Cooperative Work, CSCW '92, pages 107-114, 1992.
- [7] Douglas Engelbart, *The Mother of All Demos*. 1968.
- [8] Sylvie Noël and Jean-Marc Robert. *Empirical Study on Collaborative Writing: What Do Co-authors Do, Use, and Like?*. In Computer Supported Cooperative Work (CSCW), March 2004, Volume 13, Issue 1, pages 63-89, 2004.
- [9] Gérald Oster, Pascal Urso, Pascal Molli and Abdessamad Imine *Data consistency for P2P collaborative editing* In Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work, CSCW '06, pages 259-268, 2006.