



UNIVERSITÀ DI PISA

Corso di Laurea Magistrale in Informatica Umanistica

TESI DI LAUREA

**Visualizzazione Interattiva di Timeline
Rappresentanti l'Uso di Applicazioni Web**

Candidato: Pier Paolo Pitardi

Relatore: *Fabio Paternò*

Controrelatore: *Vincenzo Ambriola*

Anno Accademico 20014-2015

Ai miei genitori, a mia sorella, a tutta la mia famiglia e a Veronica, grazie ai quali sono la persona che sono e ai quali posso dire solo un sentito grazie.

Ad Ezio e Sonia, per avermi accolto e amato come un figlio sin dal primo giorno in cui le nostre strade si sono incrociate.

Ad Antonio, Alessio, Paolo, Manjola, Francesco, Alessandro e Martina, amici sinceri e sempre presenti, in ogni mio momento di gioia o difficoltà.

A Teresa, per essere stata con me la nonna più dolce e amabile di sempre.

Indice Generale

1. Introduzione.....	1
2. Usabilità e responsive design.....	3
3. Lo stato dell'arte.....	8
3.1 Soluzioni commerciali.....	10
3.2 Strumenti di ricerca.....	13
4. Le origini di MUSE: Web Usability Probe.....	16
5. Mobile Usability Smell Evaluator.....	23
5.1 Background.....	23
5.2 Le tecnologie alla base di MUSE.....	24
5.2.1 JQuery.....	24
5.2.2 Hammer.js.....	26
5.2.3 Almende CHAP Links Library.....	28
5.2.4 Move.js.....	29
5.2.5 HTML5.....	31
5.2.6 Twitter Bootstrap.....	31
5.2.7 Html2canvas.....	32
5.2.8 Flot.js.....	33
5.2.9 Font Awesome.....	34
5.2.10 Fontello.....	35
5.3 Le novità introdotte da MUSE.....	35
5.3.1 Il nuovo logger.....	35
5.3.2 I nuovi dati rilevabili.....	41
5.3.3 I Bad Usability Smells.....	42
5.4 I vantaggi di MUSE per il valutatore.....	45
5.4.1 La nuova interfaccia responsive.....	45
5.4.2 L'acquisizione degli screen-dump.....	54
5.4.3 Le nuove timeline.....	56
5.4.4 La nuova modalità di confronto delle timeline.....	71
5.4.5 La sezione overview.....	75
5.4.6 La visualizzazione dell'utente in movimento.....	80
6. Il test valutatore.....	83
6.1 Gli obiettivi del test.....	83
6.2 La definizione del test per il valutatore.....	84
6.3 La stesura del questionario.....	86

6.4 I risultati del test valutatore.....	88
7. Conclusioni.....	98
8. Appendice A: le domande del questionario.....	101

1. Introduzione

Con l'evoluzione continua delle tecnologie e dei dispositivi l'analisi dell'usabilità è diventata la chiave del successo delle applicazioni web.

La nascita dei dispositivi mobili ha creato nuove modalità di fruizione dei contenuti delle applicazioni, come per esempio il mobile computing.

Un tale scenario ha indotto i valutatori di usabilità ad utilizzare strumenti automatici per semplificare ed ottimizzare le attività di analisi e valutazione dell'usabilità, rendendo più semplice e veloce l'individuazione di possibili problematiche delle interfacce utente.

Con l'aumentare della complessità delle problematiche delle interfacce utente, gli strumenti automatici dedicati alla valutazione dell'usabilità sono diventati nel corso del tempo sempre più articolati, integrando in sé funzionalità sempre più raffinate e interattive.

L'obiettivo di questa tesi è la progettazione e l'implementazione di una visualizzazione interattiva dei dati memorizzati da uno strumento automatico capace di registrare il comportamento assunto dagli utenti nell'uso di applicazioni web mobili.

Il risultato del lavoro svolto consiste nella realizzazione dell'interfaccia utente del nuovo strumento realizzato.

Tale interfaccia è basata su timeline interattive che hanno introdotto nuove funzionalità e approcci innovativi in grado di facilitare ed agevolare l'analisi e la valutazione dell'usabilità di applicazioni web mobili da parte di esperti.

La tesi proposta presenterà, nel secondo capitolo, il rapporto tra l'usabilità e la tecnica del responsive design.

Nel terzo capitolo saranno illustrati i più interessanti strumenti automatici per la valutazione dell'usabilità all'attuale stato dell'arte. Più precisamente, il capitolo tratterà la descrizione di alcuni applicativi realizzati per il mercato commerciale ed altre soluzioni impiegate invece nel campo della ricerca scientifica.

Nel quinto capitolo sarà presentata in primis una panoramica delle tecnologie utilizzate per la progettazione dell'interfaccia utente dello strumento, passando in seguito alla descrizione dettagliata della novità introdotte dallo strumento stesso.

Sarà poi trattata, nel dettaglio, la struttura della nuova interfaccia e saranno ampiamente descritte tutte le nuove funzionalità interattive realizzate.

Nel sesto capitolo sarà presentata tutta la fase realizzativa di un test proposto ad alcuni valutatori di usabilità, i quali sono stati invitati ad interagire con le varie funzionalità introdotte e ad esprimere un giudizio per mezzo di un questionario elettronico. In chiusura del capitolo saranno discussi ampiamente i risultati del test, dando particolare attenzione alle critiche e ai suggerimenti forniti dai valutatori selezionati.

Infine, in appendice alla tesi saranno elencate le domande del questionario elettronico presentato.

2. Usabilità e responsive design

I dispositivi mobili (smartphone, tablet, phablet) sono stati creati con l'intento di ridurre le dimensioni degli apparati, permettendo all'utente di avere device con caratteristiche tali da poter essere messi in una tasca o in uno zaino e, allo stesso tempo, capaci di fornire agli utenti l'uso di applicazioni anche in mobilità.

La valutazione dell'usabilità, con l'arrivo dei dispositivi mobili, è diventata rilevante anche per gli stessi produttori, che hanno cambiato strategie commerciali e produttive grazie alle informazioni fornite dalle valutazioni stesse.

In particolare da tali studi è emerso come gli utenti, con schermi più grandi, riuscissero a svolgere attività sulle interfacce avendo una migliore *user experience*.

La migliore fruizione di contenuti su schermi più grandi ha indotto i produttori a creare device con display di dimensione sempre maggiore, fino ad arrivare, con i dispositivi di ultima generazione, a costruire schermi della grandezza di sei pollici.

Per avere un'idea chiara di tale tendenza da parte dei produttori, è possibile visionare la *Figura 1*, in cui è rappresentata l'evoluzione dell'*iPhone* di *Apple Inc.*



Figura 1. L'evoluzione di iPhone.

Con l'aumentare della grandezza degli schermi la qualità dei display dei dispositivi è nettamente migliorata, raggiungendo risoluzioni sempre maggiori.

Inoltre ha reso i dispositivi in grado di essere utilizzati con due orientamenti differenti, definiti con il termine tecnico di *portrait layout* e *landscape layout* (vedi *Figura 2*).

I due orientamenti citati permettono un utilizzo del dispositivo sia in senso verticale che orizzontale. Di ciò hanno tenuto conto gli sviluppatori, realizzando applicazioni con interfacce utente capaci di adattare la presentazione dei contenuti automaticamente, in base all'orientamento scelto dall'utente nell'uso del proprio device.

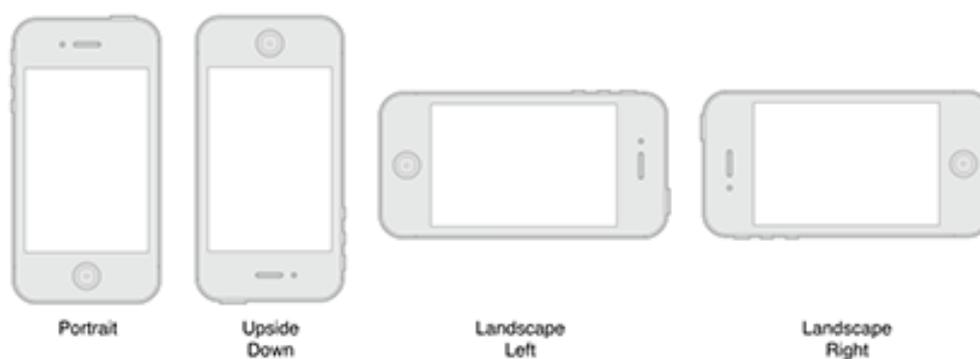


Figura 2. Portrait layout e landscape layout.

Come osservato da Raluca Budiu in [2W], l'evoluzione dei dispositivi mobili ha comportato la nascita di nuove tecniche di realizzazione delle interfacce utente, apportando grandissime novità soprattutto in ambito di design.

Una di queste è il *responsive design*, una tecnica di sviluppo web basata su un approccio che mira a creare cambiamenti dinamici alla visualizzazione delle applicazioni web, prendendo come riferimento la grandezza dello schermo e l'orientamento del dispositivo in uso.

Il responsive design è stato oggetto di un dibattito tra i professionisti del web design e gli esperti di usabilità. Tali figure professionali si sono chieste se fosse più corretto, ai fini di dare una migliore user experience all'utente, sviluppare applicazioni web realizzate con il responsive design o se, invece, fosse più conveniente lo sviluppo di applicazioni web con interfacce utente dedicate alle varie tipologie di dispositivo presenti sul mercato.

Come osservato da Raluca Budiu, è innegabile che all'utente non interessi minimamente come un'applicazione, o in particolare la sua interfaccia utente, sia stata realizzata tecnicamente. Un utente, secondo Budiu, preferisce semplicemente l'applicativo capace di garantirgli una migliore user experience, dunque l'applicazione meno caratterizzata da problematiche legate all'usabilità.

Una risposta certa a tale questione non può essere data, poiché la scelta dell'approccio è strettamente dipendente dalla quantità e dal tipo d'informazione che l'applicazione dovrà gestire.

Dunque potranno verificarsi casi in cui risulti più vantaggioso utilizzare applicazioni con interfacce dedicate ai vari tipi di dispositivo o altri in cui un approccio basato su cambiamenti dinamici come quello del responsive design possa essere ritenuto la strada più semplice e giusta da intraprendere.

Studiando la problematica da un punto di vista realizzativo, un approccio basato sul responsive design è certamente vantaggioso rispetto ad applicazioni web con interfacce utente dedicate ad ogni tipologia di device.

Il primo vantaggio riscontrabile con l'approccio responsive è senza dubbio la gestione dell'informazione. La tecnica del responsive design permette di utilizzare un unico insieme di contenuti (testuali e multimediali) all'interno dell'applicazione, proprio in virtù di quei cambiamenti dinamici di visualizzazione effettuati automaticamente dall'interfaccia.

Inoltre, l'uso del responsive design può contribuire anche al mantenimento di una coerenza visuale e funzionale, grazie ad una tipologia di visualizzazione capace di adattarsi ad ogni tipo di dispositivo.

Un altro motivo valido per ritenere la tecnica del responsive design vantaggiosa è il fatto che questa modalità di realizzazione delle interfacce basi il suo fondamento sui fogli di stile che regolano la presentazione dei contenuti dell'applicazione, assumendo come discriminante la larghezza dello schermo dei dispositivi.

Le applicazioni responsive applicano i cambiamenti dinamici al contenuto delle pagine web in relazione a questa proprietà e ad alcune misure soglia, dette *breakpoint*, che una volta rilevate permettono all'applicazione di applicare le regole di stile CSS definite per l'intervallo di riferimento.

Le applicazioni web con interfacce utente dedicate, sotto questi aspetti, comportano costi maggiori rispetto al responsive design.

Può accadere che versioni mobile e versioni desktop della stessa applicazione si basino su basi di dati diverse, ma soprattutto è possibile che l'interfaccia, se non studiata appositamente per dispositivi e orientamenti ben definiti, possa creare problemi di visualizzazione dei contenuti e generare ulteriori problemi di usabilità.

Tuttavia anche il responsive design non è certo esente da critiche in fatto di usabilità. Uno dei maggiori problemi derivati dall'uso della tecnica del responsive design è la

lunghezza delle pagine web. Tale problematica coinvolge proprio la gestione dei contenuti delle stesse pagine. Contenuti pensati per la versione desktop di un'applicazione potrebbero risultare pesanti per un dispositivo mobile. Non è un caso che gli esperti di design abbiano riscontrato migliori risultati sviluppando le interfacce con un approccio *mobile first*, ossia realizzandole partendo dalla definizione delle regole di visualizzazione dedicate ai dispositivi mobili e preoccupandosi della visualizzazione desktop solo successivamente.

Una pagina web troppo lunga, quando visualizzata su un dispositivo mobile, può comportare infatti un'eccessiva attività di scroll da parte dell'utente durante la fruizione dei contenuti.

Per risolvere tale problematica gli sviluppatori si sono tutelati introducendo nel design delle interfacce utente degli strumenti appositi capaci di raggruppare il contenuto in diverse sezioni (i cosiddetti *accordion*), dei pulsanti per effettuare un ritorno veloce all'inizio della pagina (*back-to-top*) e, per quanto riguarda la navigazione, menu, definiti *sticky*, posti sempre in primo piano anche in seguito al verificarsi di attività di scroll verticale.

Dunque il contenuto sembra essere l'oggetto chiave della tecnica del responsive design.

Ciò ha portato i designer e gli sviluppatori a porsi delle domande su come attuare cambiamenti alla sua disposizione all'interno delle pagine e su quanto tali cambiamenti possano influire sulla performance degli applicativi.

Quando si parla di performance, va considerato che il responsive design utilizzi lo stesso codice sorgente per tutti i dispositivi. Appare dunque corretto valutare il tipo di codice scritto, cercando di capire se questo sia troppo pesante su periferiche non troppo potenti o datate. I cambiamenti apportati dal responsive design avvengono lato client, attraverso il browser utilizzato dagli utenti. Quindi non è certo scontato che utilizzando un codice scritto per un dispositivo desktop si possa ottenere una performance accettabile anche su di un dispositivo mobile di entry-level.

Tornando all'usabilità, dalla nascita dei dispositivi mobili e delle nuove tecniche di sviluppo delle applicazioni web sono derivati cambiamenti radicali anche nell'approccio e negli strumenti di valutazione.

I valutatori di usabilità, vista la massiccia mole di dati sul comportamento degli utenti ottenibile dai device mobili, hanno sentito la necessità di poter ottenere

informazioni in ogni momento, cercando di effettuare analisi non più solo con dati quantitativi ma anzi con un'informazione di tipo qualitativo. Seguendo questa direzione si è resa necessaria la creazione di strumenti automatici per la valutazione dell'usabilità capaci di rappresentare i dati elaborati con differenti approcci di visualizzazione, ai fini di una semplificazione dell'attività dei valutatori.

3. Lo stato dell'arte

Come anticipato nel capitolo precedente, gli strumenti automatici per la valutazione dell'usabilità sono stati sviluppati per favorire i valutatori nella loro attività.

La valutazione dell'usabilità, come osservato da Santana et al in [1], gode di caratteristiche peculiari che hanno costituito un riferimento nello sviluppo degli strumenti di analisi automatica. Le proprietà più importanti in questione sono la posizione dei partecipanti, l'uso di task specifici, l'interazione tra utenti e valutatori e la sorgente dei dati su cui l'analisi viene a posteriori elaborata.

Prendendo in esame ciascuna di queste caratteristiche insite nella valutazione di usabilità, è possibile distinguere gli strumenti automatici a supporto del valutatore definendo i vari approcci utilizzati da tali applicativi.

Considerando la posizione degli utenti partecipanti, la valutazione dell'usabilità può essere elaborata intraprendendo due strade differenti: la valutazione effettuata *in remoto* e, come ci si può aspettare, quella effettuata *non in remoto*.

Le valutazioni in remoto permettono al valutatore e all'utente partecipante di non essere nello stesso posto e nello stesso istante nel momento in cui la valutazione stessa viene effettuata. Al contrario, con le valutazioni effettuate *non in remoto* si intende uno scenario in cui valutatore e partecipante siano presenti nello stesso posto e nello stesso momento in cui la valutazione avviene, in una sorta di ambiente controllato.

Da un punto di vista di utilizzo dell'interfaccia, la valutazione d'usabilità può invece distinguersi circa la tipologia di task che il partecipante deve compiere sull'interfaccia. I task possono essere scelti liberamente dallo stesso partecipante alla valutazione oppure essere mirati e dunque pianificati a priori dal valutatore.

Un'altra importante proprietà delle valutazioni d'usabilità è l'interazione tra partecipanti e valutatori. In questo senso le valutazioni in remoto sembrano più vantaggiose, vista la possibilità di osservare il comportamento degli utenti in tempo reale ed ottenere da questi un feedback diretto attraverso questionari elettronici o, per esempio, comunicazioni dirette e veloci inviate tramite la rete. Tuttavia, in alcuni casi l'osservazione in tempo reale può presentare delle insidie. Dipendendo strettamente dalla connessione a disposizione degli utenti, in caso di connessioni di

bassa qualità o poco potenti l'attività dei valutatori potrebbe essere problematica o, nei casi peggiori, compromessa.

La caratteristica di maggiore rilievo negli strumenti di analisi automatica dell'usabilità è, dunque, l'acquisizione di dati su cui poter elaborare a posteriori l'analisi. Per effettuare l'acquisizione dei dati, allo stato dell'arte possono essere utilizzati tre approcci differenti:

- *Acquisizione dei dati lato client*
- *Acquisizione dei dati lato server*
- *Acquisizione dei dati basata su server proxy*

L'acquisizione dei dati *lato client* può essere effettuata per mezzo di *data logger* inseriti all'interno delle stesse pagine web oppure attraverso browser modificati appositamente per tale attività (plugin realizzati ad hoc). Tra i vantaggi dell'acquisizione lato client il più rilevante è certamente l'alto dettaglio dell'informazione acquisita, a discapito però di un elevato costo in termini di risorse, dato che il data logger necessariamente deve essere presente in ogni pagina web coinvolta dal task.

La strada che vede protagonista l'acquisizione dati *lato server* prevede una tipologia d'acquisizione che avviene per mezzo d'informazioni generate da un web server durante l'accesso, da parte degli utenti, ai contenuti che lo stesso server ospita. I limiti di questa tipologia, come illustrato da Santana in [1], sono riscontrabili nell'impossibilità di riuscire a registrare informazioni circa le pagine web contenute nella cache del browser e nell'ipotetica identificazione del singolo utente, certamente più complessa nell'implementazione rispetto agli altri due approcci presentati.

Da ultimo, l'approccio di acquisizione dati basato su *server proxy* è un tipo di strada particolare rispetto agli approcci descritti precedentemente. In questo caso lo strumento di valutazione si pone come mediatore tra l'utente e le pagine web a cui viene richiesto l'accesso. Una volta effettuata dall'utente la richiesta per una specifica pagina web, questa viene in un primo momento indirizzata al server proxy che, dopo aver effettuato una richiesta http effettiva alla pagina realmente esistente, visualizza sul client dell'utente una copia della pagina richiamata su cui, nelle fasi precedenti la visualizzazione, ha iniettato il data logger che usualmente viene realizzato mediante il linguaggio di programmazione Javascript.

Per i valutatori l'utilizzo di un server proxy può risultare vantaggioso, dato che si tratta di un tipo di tecnologia in grado di permettere valutazioni di usabilità delle applicazioni web di cui questi non sono effettivamente proprietari o gestori in prima persona.

Tuttavia utilizzare un proxy non è affatto semplice, visti i possibili problemi di incompatibilità derivati dall'uso di questo approccio. I tempi di risposta del proxy stesso, la riscrittura delle URL e per esempio alcuni problemi legati alla sicurezza possono essere considerati il punto debole di questo metodo.

All'attuale stato dell'arte è possibile trovare molti strumenti automatici di valutazione dell'usabilità. Tali applicativi possono appartenere al mercato commerciale o al campo della ricerca scientifica. Nei prossimi paragrafi sarà presentata una breve panoramica di tali strumenti, tenendo conto della distinzione appena citata.

3.1 Soluzioni commerciali

Tra le soluzioni commerciali dedicate all'analisi e alla valutazione dell'usabilità, all'attuale stato dell'arte le proposte più innovative ed interessanti sono le seguenti:

- Google Analytics
- CrazyEgg
- ClickTale

Google Analytics [8W] è un tool per la valutazione di usabilità sviluppato da *Google Inc.* Questo strumento è in grado di effettuare acquisizioni di dati ed elaborare su questi un'analisi automatica.

Analytics fornisce informazioni statistiche circa numero di visitatori, pagine visualizzate, origini delle richieste, periferiche ed eventi generati dall'interfaccia. Al valutatore è richiesta una registrazione e la conseguente creazione di un account come sviluppatore per ottenere il codice Javascript in grado di attivare lo strumento. Tale codice deve essere necessariamente inserito all'interno di tutte le pagine oggetto della valutazione. L'acquisizione dati effettuata da Google Analytics è un'acquisizione lato client, quindi caratterizzata da un dettaglio informativo davvero

elevato. Lo strumento dà l'ulteriore possibilità di definire degli eventi custom personalizzabili dal valutatore, che può descriverne il nome e configurare il software indicando l'elemento dell'interfaccia su cui saranno scatenati.

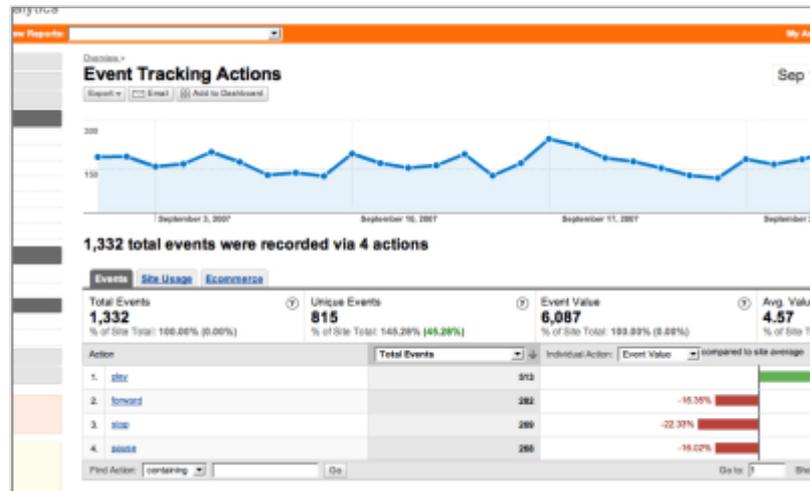


Figura 3. Esempio di visualizzazione del tracciamento degli eventi in Google Analytics.

CrazyEgg [6W] è uno strumento per l'analisi automatica dell'usabilità in remoto creato da *Crazy Egg Inc.*

Anche in questo caso siamo di fronte ad un'acquisizione dati lato client, analoga a quella effettuata da Google Analytics. Lo strumento, come quello descritto precedentemente, deve essere attivato su tutte le pagine oggetto dell'analisi tramite uno script Javascript da inserire manualmente nel codice sorgente delle stesse.

CrazyEgg fornisce ai web developer e ai valutatori di usabilità quattro funzionalità mirate, ciascuna sviluppata per analizzare diversi aspetti di usabilità. Nel dettaglio le funzionalità in questione sono le seguenti:

- *Heatmap*: visualizza, attraverso mappe di calore, i punti in cui l'utente ha cliccato o toccato lo schermo (in caso di dispositivi mobili) durante l'attività compiuta sulla pagina.
- *Scrollmap*: questa funzionalità indica quanto e fino a che punto della pagina l'utente ha compiuto attività di scrolling prima di abbandonare la stessa.

- *Overlay*: l'overlay report permette al valutatore di avere informazione dettagliata sui click effettuati su ogni singolo elemento della pagina.
- *Confetti*: la funzionalità confetti permette invece di avere informazioni sui click, organizzando e categorizzando i dati ottenuti tenendo conto delle fonti esterne di riferimento. Con questa feature è quindi possibile capire quali siano le fonti esterne in grado di garantire più visite all'applicazione oggetto della valutazione.



Figura 4. Esempio di visualizzazione heatmap di CrazyEgg.

CrazyEgg è in grado di gestire, su un solo account, l'analisi di più domini e supporta siti sicuri dotati di certificato SSL.

ClickTale è uno strumento molto valido, in grado di effettuare vere e proprie registrazioni video delle sessioni di interazione compiute sull'interfaccia dall'utente. È di per sé capace di acquisire dati lato client circa il comportamento assunto ed è caratterizzato da una visualizzazione dei dati acquisiti simile a quella di CrazyEgg.

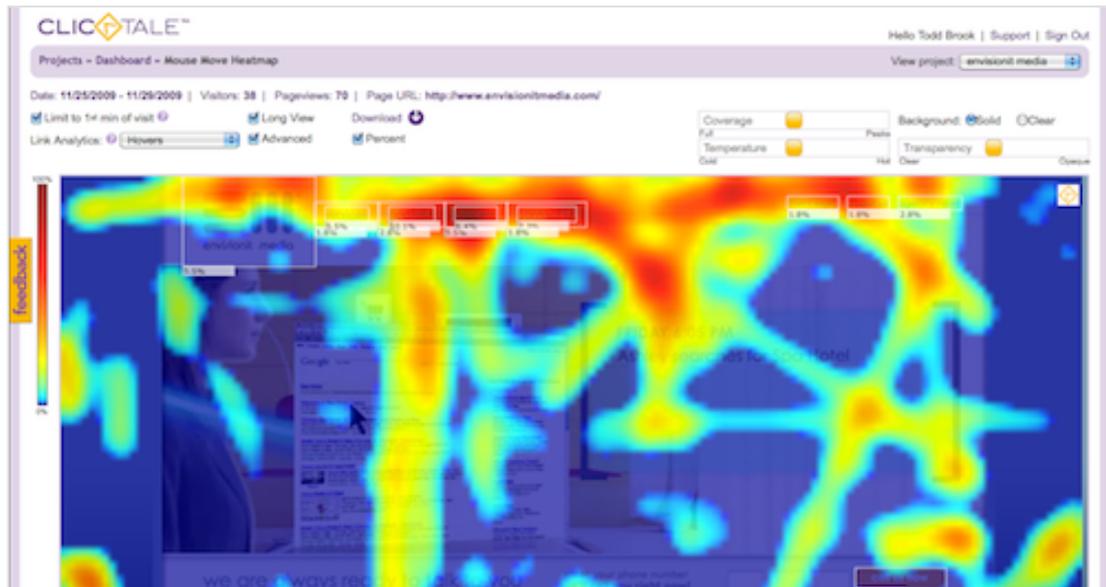


Figura 5. Esempio visualizzazione di ClickTale

Anche ClickTale utilizza mappe di calore per individuare i click degli utenti, ma soprattutto riesce ad evidenziare le aree ignorate ed i punti di distrazione caratterizzanti l'interfaccia stessa. ClickTale implementa strumenti per l'analisi dell'attività di scroll, per il conteggio delle visite ottenute dall'applicazione, per la lettura della provenienza di link esterni e per la realizzazione di grafici della tipologia *conversion funnel*.

3.2 Strumenti di ricerca

L'argomento della valutazione dell'usabilità è stato trattato anche nel campo della ricerca scientifica. In particolare segnaliamo, all'attuale stato dell'arte, i progetti di ricerca più interessanti per tale ambito:

- WELFIT - [1]
- THE BAD SMELL FINDER - [2]
- TOOL DEVELOPED BY MEDIA INFORMATICS GROUP
- UNIVERSITY OF MUNICH [16]

WELFIT è uno strumento per l'analisi automatica dell'usabilità in grado di effettuare test di valutazione in modalità remota e non remota (Santana, [1], 2014). L'acquisizione dei dati utilizzati da *WELFIT* viene effettuata lato client. È in grado di acquisire dati su ogni tipo di evento rilevabile sull'interfaccia e può effettuare report statistici, elaborando grafici rappresentanti i risultati ottenuti dall'analisi automatica. *WELFIT* è un'applicazione molto potente, ma presenta alcuni limiti tra cui il più rilevante può essere individuato nell'obbligo, da parte del valutatore di usabilità, di dover inserire manualmente in tutte le pagine interessate dal test il codice necessario all'attivazione del data logger.

THE BAD SMELL FINDER è un'applicazione per l'analisi automatica in remoto dell'usabilità che, come illustrano Grigera, Garrido e Rivero in [2], ha le sue fondamenta nel concetto di *refactoring*. Questo strumento effettua un pre-processing degli eventi, acquisiti lato client, dando la possibilità ai valutatori di redigere un report di concreti problemi di usabilità rilevati.

Il funzionamento di Bad Smell Finder si basa su tre fasi principali:

- Acquisizione lato client dei dati sugli eventi generati da utenti reali
- Processazione dei dati di log per identificare possibili problematiche di usabilità
- Ricezione lato server dei dati processati ed elaborazione di report dettagliati sulla possibile presenza di Bad Usability Smell

Quando un utente richiede un report, Bad Smell Finder processa i dati di log identificando le possibili minacce di errore. Successivamente, lo strumento realizza una visualizzazione dei bad usability smell riscontrati attraverso il frontend dell'applicazione.

Un altro applicativo interessante è nato da un progetto di ricerca condotto dal Media Informatics Group dell'Università di Monaco.

Come illustrato da Buschek et al in [16], l'applicazione colleziona dati utili sugli eventi basati sul tocco, fornendo informazioni dettagliate sull'esatta posizione in cui tali eventi si sono verificati e sugli elementi della pagina che li hanno scatenati.

Tali dati sono rilevati dallo strumento attraverso una libreria Javascript, la quale successivamente si occupa dell'invio ad una piattaforma basata sull'interazione del linguaggio PHP e di un database MySQL.

In base all'informazione rilevata lo strumento è in grado successivamente di creare, attraverso appositi metodi, modelli di possibile interazione futura da parte dell'utente. Dunque, effettua una sorta di previsione delle azioni che l'utente andrà a compiere sull'interfaccia.

Quanto alla visualizzazione dei dati, questo strumento fornisce una visualizzazione interattiva. In particolare, come si può notare in *Figura 3*, sono disponibili tre differenti visualizzazioni.

Nella visualizzazione rappresentata sul lato sinistro, sono visualizzati eventi basati sul tocco simulati dallo strumento. Tali eventi sono rappresentati con dei punti di colore rosso e sono posizionati su un'immagine rappresentante la pagina web oggetto dell'analisi.

Nella visualizzazione centrale viene invece rappresentata la distribuzione degli eventi touch all'interno della pagina. Tale distribuzione è disegnata all'interno dell'immagine per mezzo di figure ellittiche di colore verde, caratterizzate da una lieve trasparenza.

Infine, nella visualizzazione rappresentata sul lato destro sono visualizzate, per ciascun elemento della pagina, delle etichette indicanti la percentuale di possibile interazione futura da parte dell'utente. Il valore inserito nelle etichette è il risultato della previsione calcolata dallo strumento durante la fase di analisi.

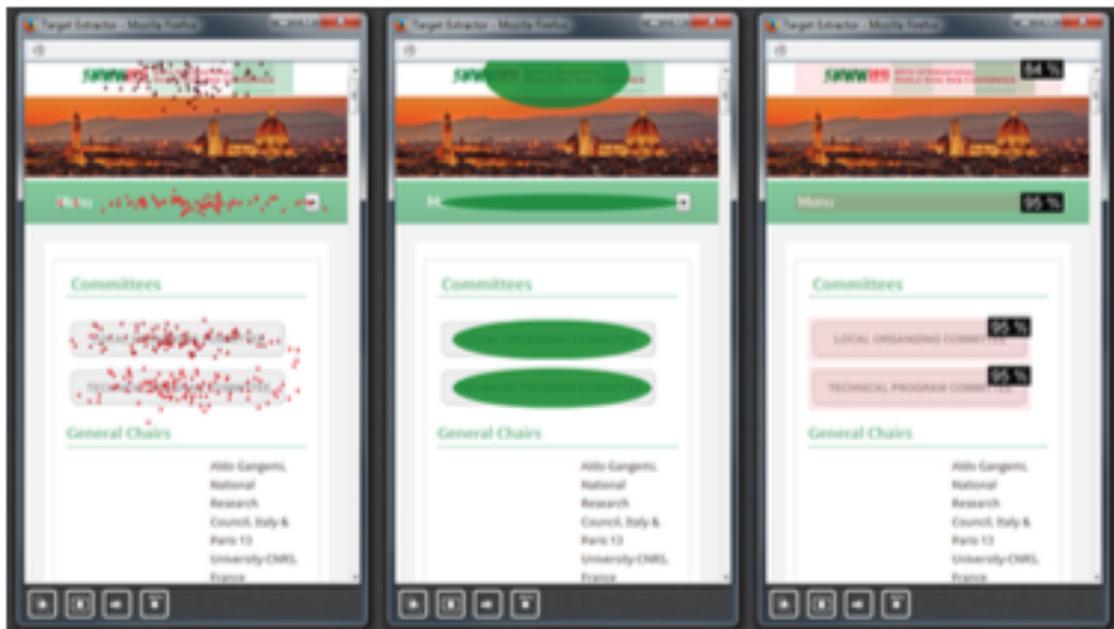


Figura 6. Esempio di visualizzazione interattiva sviluppata dal Media Informatics Group

Alla luce dell'analisi dei vari tool presentati, dopo aver analizzato pregi e difetti di tali strumenti abbiamo preso in considerazione le caratteristiche dei singoli applicativi per la progettazione e la realizzazione del nostro strumento. Tenendo conto che gli applicativi proposti in precedenza non siano in grado di effettuare confronti tra le sessioni in maniera visuale e di rendere la loro visualizzazione interattiva, questi due punti hanno costituito il focus del lavoro svolto

4. Le origini di MUSE: Web Usability Probe

Web Usability Probe è un'applicazione web per la valutazione dell'usabilità in remoto sviluppata dal laboratorio HIIS dell'Istituto di Scienze e Tecnologia dell'Informazione del CNR di Pisa.

MUSE (Mobile Usability Smell Evaluator) ha le sue radici proprio in questo strumento di valutazione, di cui può essere considerato l'erede essendone la naturale evoluzione.

Per capire lo scopo per cui MUSE è nato e le novità da esso introdotte, si ritiene necessaria una panoramica sulle funzionalità implementate all'interno del suo predecessore.

WUP è un applicativo che rientra nella categoria delle cosiddette *proxy-based application*. Si serve infatti di un server proxy per effettuare l'acquisizione della maggior parte di azioni tipiche scatenate dall'interazione compiuta dall'utente sull'interfaccia di un'applicazione web che è oggetto della valutazione stessa.

Più precisamente, in WUP l'acquisizione e la registrazione dei dati sul comportamento degli utenti coinvolgono le seguenti categorie di eventi:

- Eventi generati dalla tastiera
- Eventi generati dal mouse
- Eventi generati dal tocco (su dispositivi mobili)
- Eventi relativi all'utilizzo del GPS
- Eventi legati all'accelerometro
- Eventi dei form HTML
- Eventi semantici
- Eventi custom

Tra le categorie di eventi appena elencate, di particolare rilevanza sono gli eventi semantici, che possono essere definiti a priori dal valutatore e messi in relazione a pre-condizioni e post-condizioni in modo tale che il valutatore stesso abbia riferimenti più precisi durante la lettura e l'interpretazione delle sequenze d'interazione.

Il funzionamento di WUP ha il suo inizio con la definizione di una valutazione. La valutazione può essere definita nelle sue proprietà native, come per esempio il nome, ed è formata da attività mirate (task) che il valutatore può pianificare e creare in quantità indefinita. Tali task, all'atto del test, rappresenteranno le attività che l'utente dovrà compiere sull'interfaccia. I task possono dunque essere creati, modificati o rimossi dalla valutazione in qualsiasi momento ed hanno la peculiarità di poter essere associati ad eventi semantici ed eventi custom, garantendo al valutatore e all'applicazione stessa grande flessibilità nell'impostazione dei parametri di configurazione della stessa valutazione.

Analizzando WUP dettagliatamente, il primo aspetto da considerare è la sua architettura, strutturata attraverso quattro componenti principali:

1. Logger degli eventi lato client
2. Server Proxy
3. Applicazione per l'elaborazione ed analisi dei dati
4. Piattaforma di memorizzazione dei dati

- *Il logger:*

La libreria incaricata della registrazione degli eventi scatenati sull'interfaccia è stata sviluppata con il linguaggio di programmazione Javascript ed è in grado di rilevare tutte le azioni tipiche che si effettuano navigando una pagina web. Questa libreria effettua la registrazione su database di tutti gli eventi rilevati attraverso la tecnologia AJAX, dedicata alla memorizzazione dei dati rilevati su un database realizzato con piattaforma MySQL.

- *Il server proxy:*

Tra i vari approcci possibili per effettuare l'inserimento del data logger all'interno delle pagine, quello utilizzato all'interno di WUP è un approccio che implica un'acquisizione dati basata su server proxy. Utilizzare un server proxy, come anticipato in precedenza nel terzo capitolo di questa tesi, ha permesso di effettuare valutazioni di usabilità non obbligando i valutatori e i partecipanti ad essere presenti nello stesso posto e nello stesso momento. Di conseguenza, da un punto di vista della qualità delle valutazioni, tale

approccio favorisce un'acquisizione dei dati durante l'utilizzo quotidiano tipico dei dispositivi, fornendo dunque un alto livello di realismo nell'informazione rilevata. Tuttavia utilizzare un server proxy non è certo semplice, viste le problematiche legate alla sicurezza e alla fase di riscrittura delle URL. Se infatti è vero che WUP, attraverso il proxy, faccia da mediatore tra utenti e valutatori, è altrettanto vero che le problematiche in questione abbiano comportato un duro lavoro per ridurre al minimo le incompatibilità riscontrate in fase di sviluppo. Uno dei problemi principali affrontati nello sviluppo di WUP è stata la S.O.P (*Same Origin Policy*). La Same Origin Policy è una misura di sicurezza lato client presente ormai in tutti i browser più avanzati. Essa garantisce che le varie richieste remote verso una risorsa web siano effettuate dallo stesso dominio di appartenenza dell'applicazione che ne faccia richiesta. In questo modo non si permette l'accesso ai metodi delle varie librerie nel caso in cui questi siano richiamati da origini diverse da quella del dominio di riferimento dell'applicazione, impedendo così la possibilità che script esterni, o eventuale codice malevolo, possano effettuare attività dannose per l'applicazione. Appoggiarsi ad un server proxy ha quindi permesso a WUP di catturare il codice sorgente delle pagine oggetto del test e di far risultare le varie richieste http come provenienti da un'unica origine, ossia il server proxy stesso. È infatti aggirando la S.O.P che siamo riusciti, con WUP, ad iniettare dinamicamente nel codice sorgente delle pagine ricreate dal proxy lo script del data logger, rendendo possibile la registrazione degli eventi e delle touch gesture scatenate sull'interfaccia durante l'interazione dell'utente con la stessa.

- *Il modello di dati:*

Per comprendere le funzionalità caratteristiche di WUP è necessario conoscere, con maggiore dettaglio, il modello di dati dell'applicazione.

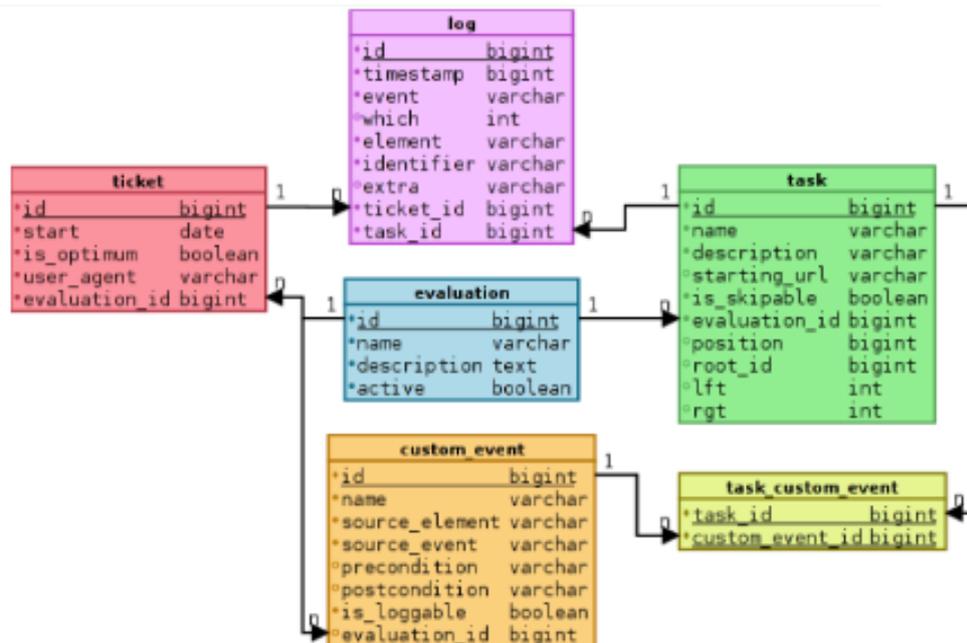


Figura 7. Modello di dati di WUP.

Il modello di dati di WUP è costituito da un'entità centrale, definita col nome di *evaluation*, che è in relazione con tutte le altre entità del database.

L'entità *task* contiene informazioni circa i vari task che compongono la valutazione ed è a sua volta in relazione con l'entità *log*, designata alla memorizzazione dei dati attinenti gli eventi registrati durante uno specifico task. Questa entità è in relazione con l'entità *ticket*, che identifica ogni singola sessione di test effettuata dagli utenti e costituisce la base di dati da cui vengono generate le singole timeline.

L'entità *custom_event* gestisce invece le informazioni riguardanti gli eventi custom che il valutatore può definire in fase di preparazione della valutazione ed è in relazione con le entità *task_custom_event* e *ticket*. La relazione con *ticket* è stata definita per collegare gli eventi custom alla sessione d'interazione in cui si sono verificati.

Infine, l'entità *task_custom_event* gestisce i dati riguardanti i custom event dei singoli task ed è in relazione con le entità *custom_event* e *task*. Sulla base di questo modello di dati WUP è stato sviluppato con delle

funzionalità ben precise, che spaziano dalla definizione delle valutazioni e delle relative componenti, fino all'elaborazione e alla visualizzazione dei dati registrati.

Le singole entità dunque sono interamente gestibili dall'applicazione, che è in grado di creare, modificare e in caso di necessità rimuovere valutazioni, task e log degli eventi memorizzati.

- *Funzionalità e visualizzazione:*

Osservando lo scenario d'uso di WUP da un punto di vista dell'utente, quando un utente compie un test ha a sua disposizione un pannello dedicato in cui sono stati inseriti i pulsanti per iniziare oppure saltare un task per passare a quello successivo. Sullo stesso pannello sono visualizzate anche le istruzioni, redatte a priori dal valutatore durante la definizione del task stesso, attraverso cui l'utente può avere un riferimento alle azioni che dovrà intraprendere durante lo svolgimento dell'attività. Da un punto di vista della visualizzazione, i dati acquisiti da WUP vengono visualizzati per mezzo di timeline poste a confronto con una timeline ottimale eseguita dal valutatore prima dell'inizio dei test. La timeline ottimale rappresenta il comportamento più idoneo per lo svolgimento del task per cui è stata definita. Ogni task è dunque caratterizzato dalla presenza di una timeline ottimale di riferimento.

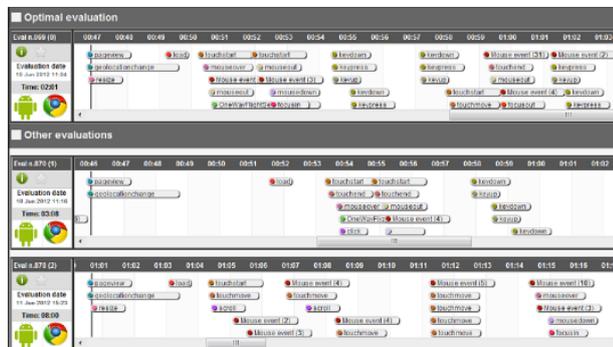


Figura 8: Esempio di visualizzazione delle timeline di WUP.

Gli eventi all'interno delle timeline di WUP sono invece visualizzati tramite etichette.

Ad ogni categoria di evento è stato associato un colore diverso, per non confondere il valutatore nella lettura della sequenza di interazione. Ogni evento fornisce informazioni utili al valutatore, come per esempio il tempo esatto in cui si è verificato, l'elemento su cui è stato rilevato ed altre informazioni variabili a seconda della tipologia di cui fa parte.

Le timeline permettono inoltre al valutatore di interagire con gli eventi. È possibile infatti cercare eventi o addirittura applicare dei filtri di selezione, rendendoli così visibili o meno all'interno della timeline in cui sono visualizzati.

Alle timeline è stata poi implementata una funzionalità relativa alla visualizzazione delle sequenze di navigazione effettuate dall'utente. Attraverso questa feature è possibile controllare all'interno della sequenza di navigazione le pagine visualizzate dall'utente stesso, fornendo anche informazioni circa il tempo utile passato dall'utente nella consultazione della pagina.

Un'altra importante funzionalità, associata a quella appena descritta, è la realizzazione di screenshot relativi agli eventi semantici e agli eventi custom. Via via che l'utente naviga di pagina in pagina, il sistema effettua uno screenshot per ciascuna delle pagine visualizzate.

Concludendo, l'ultima importante novità portata da WUP è stata l'applicazione dell'algoritmo SAM (*Sequence Align Method*) alla sequenza di eventi registrati durante il test.

L'algoritmo SAM interpreta ogni evento come un oggetto indipendente. Quest'algoritmo valuta la somiglianza tra due sequenze, quindi tra due timeline, analizzando e calcolando quante operazioni siano necessarie per trasformarle in due sequenze identiche. Le operazioni ammesse dalla SAM sono il riordinamento, l'inserimento e la rimozione di eventi. Minore è il valore risultante della SAM maggiore sarà la somiglianza tra le due sequenze. Va specificato che il calcolo della SAM è un'operazione di calcolo pesato. Ciò comporta che ad ogni tipologia di operazione ammessa (riordino, inserimento o rimozione) possa essere dato un peso in ragione di quali siano le finalità dell'analisi per cui la SAM viene calcolata.

Per concludere, saranno elencati i limiti di WUP che hanno portato alla realizzazione del nuovo strumento di valutazione dell'usabilità che abbiamo realizzato:

- Mancanza di una sezione riassuntiva dell'andamento generale dei singoli task
- Mancanza di una procedura di confronto più immediata e semplificata rispetto all'approccio usato con l'algoritmo SAM
- Mancanza di una procedura in grado di individuare problemi di usabilità nelle sequenza d'interazione
- Mancanza di una visualizzazione dei problemi di usabilità all'interno della sequenza di azioni rappresentata
- Basso livello d'interattività delle timeline
- Mancanza di dati qualitativi a supporto dell'analisi delle sequenze

5. Mobile Usability Smell Evaluator

5.1 Background

MUSE è un'applicazione web dedicata alla realizzazione di valutazioni remote di usabilità. Proprio per la sua natura MUSE, come il progetto da cui ha origine, permette di intraprendere test utente e valutazioni dell'usabilità con un approccio che rende indipendenti l'uno dall'altro utente e valutatore. Considerando quest'aspetto ed associandolo alla rivoluzione tecnologica e sociale apportata dai dispositivi mobili, ad oggi è riscontrabile che la mole di dati prodotta dalle applicazioni web sia aumentata a dismisura, fornendo così per analisi e valutazioni dell'usabilità una mole di dati utili di grandissimo volume.

Data la grandezza del volume di dati, è inevitabile che all'interno di questo grande agglomerato d'informazione sia presente un'ingente quantità di rumore, la quale ha portato di conseguenza gli esperti della valutazione dell'usabilità a ritenere molti dei dati a loro disposizione non propriamente rilevanti.

Rilevare soltanto l'informazione utile alle valutazioni di usabilità è diventata quindi la sfida da intraprendere per ottenere dei feedback realistici sul comportamento degli utenti durante l'utilizzo dei dispositivi, sul generale contesto d'uso e per giunta su pregi e difetti delle applicazioni.

Per rimediare a questa problematica, i valutatori e gli sviluppatori di applicazioni hanno cercato nel corso degli anni di creare o adeguare gli strumenti automatici di valutazione come MUSE alla continua evoluzione dei dispositivi e alla ricerca intelligente dei soli dati necessari alla valutazione automatica dell'usabilità. Possono essere considerate un buon esempio di questa tendenza le moderne tecniche di Data Mining che, attraverso algoritmi di clustering e classificazione, hanno di certo semplificato lo scenario proprio perché nate per l'analisi di sistemi complessi.

Tuttavia, il Data Mining non è di certo l'unica via per la risoluzione della problematica.

Anche la visualizzazione dei dati è certamente un nodo cruciale per un'analisi e un'interpretazione chiara e coerente dei dati sul comportamento degli utenti. L'avvento dei nuovi dispositivi mobili sul mercato ha portato grossi cambiamenti

anche in quest'ambito, come testimonia di certo l'evoluzione delle tecniche e degli strumenti che il web design ha introdotto nel corso degli ultimi anni.

MUSE, da questo punto di vista, è sicuramente uno strumento molto flessibile. In fase di sviluppo è stata valutata l'eventuale possibilità di progettare funzionalità basate su algoritmi di Data Mining. Tuttavia, ci è sembrato molto più produttivo in un primo momento far sì che il valutatore avesse una presentazione dei dati funzionale alla facile interpretazione degli stessi, introducendo un'interfaccia completamente ridisegnata e nuove funzionalità interattive dedicate alla generazione di statistiche e grafici chiari e coerenti in grado di fornire una prima indicazione sull'andamento dei singoli task.

In questo capitolo verrà analizzato MUSE nelle sue componenti principali e verranno presentate le varie novità apportate nello strumento con lo scopo di ottenere dalle valutazioni automatiche effettuate dal valutatore una maggiore immediatezza nella presentazione dello scenario d'uso generato dagli utenti e una maggiore precisione nell'identificare il loro comportamento, rendendo il processo valutativo più semplice sotto tutti i punti di vista.

5.2 Le tecnologie alla base di MUSE

Per la realizzazione di MUSE è stato necessario aggiornare e/o introdurre tecnologie specifiche in grado di garantire l'effettivo raggiungimento degli obiettivi del progetto.

Ogni fase di funzionamento è quindi basata su librerie e plugin, per lo più scritti col linguaggio di programmazione Javascript, che hanno permesso di realizzare MUSE e di oltrepassare i limiti del suo predecessore (WUP).

5.2.1 JQUERY

Nato nel 2006 dall'idea di John Resig, JQUERY è un framework Javascript sviluppato con lo scopo di snellire il codice prodotto dagli sviluppatori e creare un sistema di estensione delle funzionalità native dei browser in grado di favorire la compatibilità tra le varie librerie.

Con il grande successo ottenuto nel corso degli anni, JQUERY è ormai divenuto una base fondamentale dello sviluppo delle applicazioni web.

Questa libreria offre metodi e funzionalità progettate per effettuare la manipolazione degli elementi delle interfacce, la navigazione trasversale del Document Object Model e la rilevazione dei più comuni eventi generati dall'interazione degli utenti con le interfacce.

La rilevazione degli eventi è supportata da specifiche funzioni attivabili per ogni tipologia di evento identificabile.

JQUERY è stato utilizzato all'interno di MUSE per la rilevazione delle seguenti categorie di eventi:

- Eventi rilevabili sui form HTML
- Eventi rilevabili dalla tastiera
- Eventi rilevabili dal mouse
- Eventi semantici
- Eventi semantici custom
- Touch Gestures (se device è un dispositivo mobile)

Oltre alla rilevazione delle azioni appena elencate, JQUERY mette a disposizione tra le sue funzionalità anche metodi per effettuare richieste http asincrone per mezzo della tecnica AJAX.

Da sottolineare è il fatto che JQUERY abbia ridotto moltissimo la quantità di codice necessaria per l'implementazione di richieste AJAX. La grande flessibilità fornita da tali metodi ha reso nel tempo la libreria di Resig quasi uno standard per ciò che concerne l'invio di richieste asincrone, rendendo tale tecnica molto più semplice rispetto alla sua forma nativa.

I metodi AJAX di JQUERY sono stati utilizzati in MUSE per instaurare la comunicazione tra l'applicazione e il suo database di riferimento, permettendo la memorizzazione dati relativi agli eventi su piattaforma MySQL.

5.2.2 HAMMER.JS

HAMMER.JS [15W] è una libreria open source in grado di rilevare eventi creati da dispositivi basati sul tocco.

HAMMER.JS è una libreria molto utile, poiché rispetto alle possibilità fornite da JQuery rende possibile la rilevazione delle azioni effettuate su dispositivi touch con la particolare peculiarità di riuscire a riconoscere anche gli eventi scatenati da un'interazione compiuta con più dita. Tali eventi sono conosciuti col nome di multi-touch gestures.

La libreria viene rilasciata in due versioni principali: una minificata per la produzione e una non minificata per l'attività di sviluppo. La versione utilizzata in MUSE è la 2.0.4.

Questa versione è molto differente da quella rilasciata nella prima release di HAMMER.JS.

È infatti completamente riscritta e supporta i più recenti browser (mobile e desktop) che fanno uso della proprietà CSS touch-action.

Inoltre, per garantire maggiore flessibilità, è stata inserita la possibilità di utilizzare contemporaneamente più istanze della libreria, in modo tale da poter utilizzare la stessa anche con un approccio multi-utente.

Il funzionamento di HAMMER.JS si basa su alcuni metodi basilari che possono essere estesi per ottenere un maggior controllo sulla rilevazione delle gesture.

Viene infatti dichiarata, all'interno del codice, l'istanza di un oggetto Javascript attraverso il metodo *Manager* della libreria. Tale metodo, per essere implementato, richiede come parametri necessari al proprio funzionamento il selettore JQUERY con cui l'elemento del DOM viene identificato e una serie di opzioni di configurazione.

Ogni tipo di evento può essere rilevato attraverso funzioni specifiche della libreria che mettono in ascolto il browser sulle gesture per cui dovrà attivare comportamenti specifici. Per fare un esempio, la gesture *pan* avrà una specifica funzione di ascolto denominata *Pan*, così come *pinch* ne avrà una una col nome di *Pinch*.

Ogni gesture ha dunque un metodo dedicato per consentire la propria rilevazione all'interno dell'interfaccia.

Una volta definite le funzioni di ascolto, queste devono essere aggiunte all'istanza del Manager creata in precedenza, facendo sì che con il metodo *on* della libreria (derivato da JQUERY) sia possibile definire il comportamento specifico da intraprendere per ogni gesture rilevata durante l'interazione con l'interfaccia.

Da un punto di vista del codice ecco qui un esempio d'implementazione della libreria HAMMER.JS:

```
// Creazione istanza Hammer Manager

var mc = new Hammer.Manager(myElement, myOptions);

// Aggiunta degli eventi scelti al manager di Hammer
mc.add( new Hammer.Pan({ direction: Hammer.DIRECTION_ALL,
threshold: 0 }) );
mc.add( new Hammer.Tap({ event: 'quadrupletap', taps: 4})
);

// Funzioni di ascolto degli eventi pan e tap
mc.on("pan", handlePan);
mc.on("quadrupletap", handleTaps);

/* Funzioni callback avviate dopo la rilevazione degli
eventi pan e tap */
function handlePan {
    alert('This is a pan !!!');
}
function handleTaps {
    alert('You made some taps !!!');
}
```

Una delle grandi potenzialità di questa libreria è la creazione di eventi custom composti. Con HAMMER.JS è possibile definire eventi attraverso la definizione di sequenze di touch gesture appartenenti a diverse tipologie. Per esempio è possibile

creare un evento *tripletap* definendolo come la sequenza di tre tap consecutivi o, in alternativa, come il susseguirsi della gesture *doubletap* seguita da un singolo tap. Per compiere quest'operazione HAMMER.JS si serve di due metodi definiti all'interno della libreria e creati appositamente per questa funzionalità: *recognizeWith()* e *requireFailure()*. Con il primo è possibile indicare ad HAMMER.JS di effettuare il riconoscimento di una gesture tramite il metodo dedicato alla rilevazione degli eventi base di cui tale gesture è composta. Il secondo invece (*requireFailure()*), disattiva le funzioni di ascolto di alcune gesture durante la rilevazione di eventi compositi, garantendo grandissima flessibilità allo sviluppatore che è così in grado di definire il comportamento della libreria con un livello di dettaglio molto alto, evitando problemi di compatibilità o l'eventuale sovrapposizione di metodi non desiderati durante il processo di rilevazione.

All'interno di MUSE HAMMER.JS è stato utilizzato per estendere la gamma di eventi rilevabili dall'applicazione. Questa libreria ha permesso a MUSE di rilevare le gesture multi-touch, con il vantaggio di avere una qualità dell'informazione fornita molto più alta, dato il dettaglio più elevato fornito dalla libreria rispetto ai metodi di rilevazione tradizionali.

5.2.3 ALMENDE CHAP LINKS LIBRARY

ALMENDE CHAP LINKS LIBRARY [16W] è una libreria Javascript per la visualizzazione di grafi, reti e timeline temporali all'interno di pagine web.

La libreria è stata sviluppata da Almende, società facente parte del Collective Human Agent Paradigm.

Tra i vari strumenti messi a disposizione da questa libreria, il nostro interesse durante le fase di sviluppo di MUSE si è focalizzato sulla componente delle timeline temporali.

Volendo dotare MUSE di una visualizzazione di questo tipo, la nostra decisione è stata quella di utilizzare questa libreria in quanto ritenuta molto più aggiornata, flessibile ed avanzata rispetto a quella precedentemente utilizzata dal predecessore di MUSE (WUP, ndr).

La scelta di tale libreria è avvenuta durante la progettazione della parte dell'interfaccia di MUSE dedicata ai valutatori (backend),

Le timeline di CHAP Links Library garantiscono un'ottima visualizzazione degli elementi su ogni tipo di device .

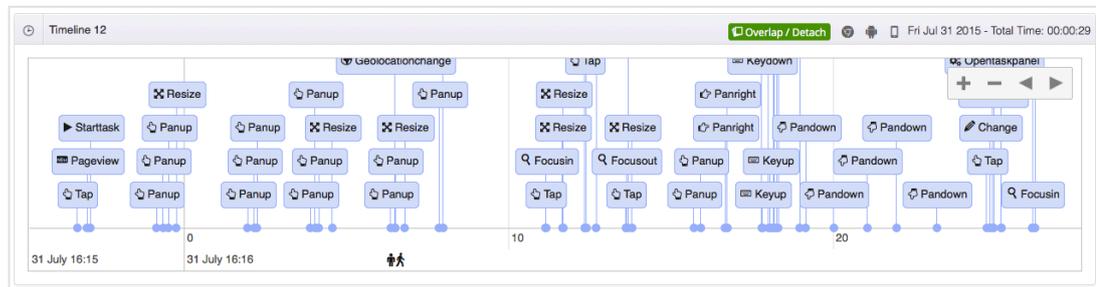


Figura 9. Esempio delle timeline di MUSE.

Le timeline di CHAP LINKS LIBRARY visualizzano eventi in sequenza temporale, con la possibilità di inserire eventi caratterizzati da una singola data o durevoli e quindi appartenenti a un determinato intervallo di tempo. Attraverso la libreria gli eventi possono essere creati, modificati e cancellati, con l'ulteriore potenzialità di avere una scala temporale dinamica che automaticamente è in grado di riposizionare gli eventi sulla timeline in base al grado di zoom scelto dal valutatore.

Un altro aspetto che ci ha convinti nell'utilizzare CHAPS LINKS LIBRARY in MUSE è l'alto grado di interazione che questa la libreria mette a disposizione dei suoi fruitori.

CHAP LINKS LIBRARY, da questo punto di vista, è risultata molto flessibile. Ha permesso infatti di effettuare manipolazioni delle timeline, come lo spostamento e l'attività di zoom, mediante comandi appositi o attività di dragging. In aggiunta, l'uso delle timeline di Almende ha consentito di definire metodi e funzioni personalizzate attivabili tramite interazione diretta (attraverso eventi click o un tap) sui box rappresentanti i singoli eventi. Abbiamo così potuto visualizzare i dati delle single azioni rilevate e perfino effettuare, a nostro gusto, dei cambiamenti di stile che hanno migliorato la lettura delle sequenze rappresentate.

Per quanto riguarda il numero di eventi gestibili dalle timeline di CHAP LINKS LIBRARY, dalla documentazione ufficiale è emerso che queste siano in grado di visualizzare fino a mille eventi per ciascuna timeline senza subire un calo di performance. Nonostante ciò, non abbiamo rilevato alcun limite al numero massimo di eventi rappresentabili.

5.2.4 MOVE.JS:

MOVE.JS [11W] è una libreria Javascript che permette di effettuare animazioni CSS per spostare, all'interno dell'area visualizzabile, gli elementi presenti nelle interfacce web.

Essendo una libreria sviluppata come estensione di JQUERY, il funzionamento di MOVE.JS è analogo a quello dei classici plugin realizzati per il framework sviluppato da Resig.

MOVE.JS richiede come parametro un selettore JQUERY necessario ad identificare l'elemento target dell'animazione, sul quale successivamente la libreria richiama i vari metodi dedicati allo spostamento.

All'interno di MUSE, per alcune operazioni di confronto tra le varie timeline, sono stati utilizzati i seguenti metodi di MOVE.JS:

- `.add(prop, value)`: è un metodo grazie al quale MOVE.JS è in grado di incrementare una proprietà CSS dell'elemento. Accetta come parametro una stringa rappresentante la proprietà da incrementare (per esempio *margin-left*) e un valore numerico che indica la quantità di pixel da aggiungere a tale proprietà.
- `.sub(prop, value)`: è un metodo con cui MOVE.JS decrementa il valore di una specifica proprietà CSS dell'elemento. Accetta come parametro, in maniera analoga al metodo precedentemente descritto, una stringa che identifica la proprietà da incrementare (per esempio *margin-right*) e un valore numerico che indica la quantità di pixel da detrarre a tale proprietà
- `.translate(x, y) / .move(x, y)`: sono due metodi equivalenti che applicano una traslazione sugli assi x e/o y, spostando l'elemento in un sistema di coordinate di riferimento di tipo cartesiano. Richiede come parametri due interi che rispettivamente rappresentano il valore dell'ascissa x e dell'ordinata y.

Particolare attenzione è certamente da dedicare ai metodi *move/translate*, dato che tali metodi sono utilizzati per la definizione delle procedure *overlap timeline* e *detach timeline* che illustreremo più avanti dettagliatamente. Le due procedure in questione sono due procedure dedicate al confronto visuale delle timeline. La prima, in particolare, fa sì che la timeline oggetto dell'overlap sia spostata esattamente sopra la timeline ottimale, così da favorirne un confronto anche grazie al cambiamento dinamico, tramite JQUERY, del coefficiente di opacità definito dai CSS dell'elemento. Per concludere, *detach timeline* non fa altro che ripristinare la situazione iniziale precedente l'overlap.

5.2.5 HTML5 API

Le HTML5 API [9W] sono un insieme di procedure che forniscono un set di strumenti specifici per la realizzazione di determinati compiti all'interno di un'applicazione web.

La documentazione e le linee guida per l'implementazione di queste procedure sono state descritte da una specifica rilasciata dal consorzio W3C. A tale specifica si sono attenuti i maggiori produttori di browser, integrando le stesse nei loro applicativi dedicati alla navigazione di pagine web.

Divenuto ormai uno standard per la realizzazione di pagine web, HTML5 ha introdotto con le proprie API un vero e proprio aumento significativo delle potenzialità dei browser moderni, rendendoli sempre più innovativi e interessanti, specialmente per gli sviluppatori di applicazioni web.

In particolare, da un'attenta lettura della documentazione rilasciata con la specifica del W3C, di grande interesse per il funzionamento di MUSE sono state le seguenti procedure introdotte da tali API:

- Gestione della grafica libera (tag `<canvas>`)
- Gestione della grafica vettoriale (tag `<svg>`)
- Memorizzazione di dati direttamente nel browser (WebSQL e LocalStorage)
- Funzionalità Javascript asincrone e parallele (Web Workers)
- Informazioni sulla posizione (HTML5 GeoLocation API)

5.2.6 *TWITTER BOOTSTRAP*

TWITTER BOOTSTRAP [17W] è un framework per la realizzazione di interfacce utente. È basato sulle tecnologie HTML5/CSS/Javascript/LESS/SASS.

Sviluppato da Mark Otto e Jacob Thornton, dipendenti di Twitter, Bootstrap è nato per supportare la creazione d'interfacce web con un approccio orientato alla coerenza e alla semplicità, vista la grande confusione generata dall'enorme quantità di librerie Javascript e CSS sviluppate in piena autonomia da aziende e web developer.

BOOTSTRAP è un framework molto articolato ed è composto da più componenti. Il framework di Twitter permette l'implementazione di svariate tipologie di strumenti, utilissimi per la realizzazione di interfacce web con un contenuto dispendio di tempi di lavoro.

Grazie alla grande visibilità di cui gode Twitter, BOOTSTRAP è diventato negli ultimi anni quasi uno standard per lo sviluppo di interfacce utente.

5.2.7 *HTML2CANVAS*

HTML2CANVAS [10W] è una libreria Javascript che permette la creazione di screenshot rappresentanti intere pagine web o singole parti di esse.

La libreria viene rilasciata in due possibili versioni: una standalone ed una realizzata come plugin dedicato all'estensione delle funzionalità native della libreria JQuery.

La scelta di questa libreria è stata fatta in seguito alla valutazione di altre soluzioni presenti sul web in grado di espletare le stesse funzionalità. Molte delle altre soluzioni trovate adottavano un approccio simile a questa libreria, differenziandosi specialmente per il fatto di essere soluzioni commerciali e quindi non open-source. L'uso di tali librerie, data la propria natura commerciale, ci avrebbe obbligato a fare uso di server proxy privati ed esterni, non permettendo così la completa gestione e personalizzazione della procedura di screen-dump.

HTML2CANVAS, nell'affrontare la creazione di screenshot delle pagine web, ha dato soluzione a queste due problematiche principali.

Html2canvas, dunque, è una libreria open-source ed ha una flessibilità notevole.

Differenzia il suo funzionamento in base alla posizione della pagina o dell'elemento di cui deve essere effettuato lo screen-dump. Nel caso in cui una pagina sia appartenente allo stesso dominio dell'applicazione, la libreria dà la possibilità inserire semplicemente il codice Javascript necessario alla sua attivazione, passando alla funzione incaricata un selettore dell'elemento target o dell'intero body. Nel caso in cui la pagina sia invece posizionata su un dominio differente a quello dell'applicazione di riferimento, la libreria permette di impostare, nei suoi parametri di configurazione, un server proxy attraverso cui è possibile acquisire immagini di pagine web esterne.

Da un punto di vista della codifica delle immagini, la libreria utilizza l'elemento canvas di HTML5 per effettuare il rendering dell'immagine acquisita, permettendo la memorizzazione su file PNG e JPG.

5.2.8 FLOT.JS

FLOT.JS [12W] è una libreria Javascript dedicata al disegno e alla realizzazione di grafici su pagine web. Viene rilasciata sotto forma di plugin per l'estensione delle funzionalità e delle procedure di JQuery.

FLOT.JS è una libreria di grande qualità da un punto di vista della visualizzazione e dell'occupazione di memoria. Queste sue peculiarità permettono di utilizzarla con lo scopo di creare grafici interattivi di aspetto davvero curato fin nei minimi particolari, senza intaccare minimamente la performance del sistema. La libreria dà la possibilità di scegliere, in fase di configurazione, il tipo di grafico necessario per la rappresentazione dell'informazione. I grafici realizzati da Flot.js garantiscono un alto livello d'interattività.

Colori, posizioni e formato degli elementi sono completamente personalizzabili. Attraverso le opzioni di configurazione della libreria è possibile applicare filtri automatici che, in base a una soglia definita, nascondono dai grafici implementati tutti i dati relativi all'intervallo di valori considerati trascurabili. Questa potenzialità della libreria, da un punto di vista della visualizzazione

dell'informazione, garantisce una maggiore comprensione dei dati rappresentati.

FLOT.JS, nei primi test effettuati in laboratorio, si è rivelata davvero molto flessibile. Ogni tipo di grafico, con tutte le sue componenti e procedure, è stato realizzato sotto forma di plugin aggiuntivo, dandoci la possibilità di utilizzare solamente il plugin necessario alla rappresentazione dei dati per mezzo della tipologia di grafico scelta.

Quanto alle tipologie di grafico, la gamma offerta da FLOT.JS. è davvero varia e mette a disposizione dello sviluppatore i seguenti tipi di rappresentazione:

- Linee
- Grafici a torta
- Istogrammi
- Grafici con aggiornamento dati in tempo reale
- Grafici a curve
- Grafici direzionali
- Grafici Gantt
- Grafici a piramide
- Istogrammi

5.2.9 FONT AWESOME

FONT AWESOME [13W] è un set di icone vettoriali scalabili progettate appositamente per essere inserite all'interno di pagine web. Le icone messe a disposizione da FONT AWESOME sono utilizzabili gratuitamente e costituiscono uno dei migliori web font reperibili sulla rete. Il set comprende cinquecentottantacinque immagini, con la caratteristica peculiare di poter essere scalata all'infinito senza nessuna perdita di qualità. Le icone di FONT AWESOME sono completamente personalizzabili, permettendo al web designer di cambiare, a proprio gusto o per necessità, colore, grandezza e ombra applicata alle stesse icone. Tra le grandi potenzialità di FONT AWESOME, di grande importanza è la compatibilità di tali icone con la maggior parte degli screen reader presenti sul

mercato. Nate in un primo momento a supporto del framework Twitter Bootstrap, le icone di FONT AWESOME sono state rilasciate successivamente in versione standalone, permettendo ai web designer di implementarle in qualunque tipo di progetto. Questo set di icone supporta anche i nuovi schermi conosciuti col nome di retina display.

La versione di FONT AWESOME utilizzata all'interno di MUSE è la 4.0.4.

5.2.10 FONTELLO

FONTELLO [14W] è uno strumento, presente sul web, dedicato alla generazione di set di icone personalizzabili. FONTELLO permette di fondere più set di immagini vettoriali scalabili in un unico set personalizzato. In MUSE è stato utilizzato per creare un set comprendente tutte le icone di FONT AWESOME e le icone dei browser meno conosciuti, presenti in altri set d'icone reperibili in rete.

5.3 Le novità introdotte da MUSE

Il primo passo intrapreso nello sviluppo di MUSE è stata un'attenta analisi del suo predecessore, WUP. Analizzando attentamente WUP, abbiamo riscontrato alcuni limiti che abbiamo deciso di superare con la realizzazione del nuovo strumento. Le modifiche e le novità apportate da MUSE hanno coinvolto vari ambiti delle funzionalità di WUP, a partire dalla raccolta dei dati fino ad arrivare alla loro elaborazione e successiva visualizzazione. Essendo MUSE un'applicazione web, uno dei primi obiettivi che ci siamo posti è stato quello di adeguare lo strumento alle nuove potenzialità offerte dalle librerie che negli ultimi anni sono riuscite a dimostrarsi più efficienti e compatibili in un contesto multiplatforma. In questo capitolo saranno illustrate, nel dettaglio, le decisioni che hanno portato alla realizzazione dello strumento, mettendo in risalto tutte le caratteristiche che, in alcuni casi, sono state rimosse o modificate perché ritenute non più al passo con

l'evoluzione delle risorse disponibili in rete e delle tecnologie. Inoltre sarà presentata una minuziosa descrizione delle nuove funzionalità introdotte.

5.3.1 *Il nuovo logger*

La raccolta dei dati è un aspetto di grande rilevanza per gli applicativi in grado di effettuare valutazioni remote di usabilità. Raccogliere dati sul comportamento degli utenti durante lo scenario d'uso quotidiano delle applicazioni web è fondamentale per una valida e corretta valutazione dell'usabilità. Tenendo in considerazione l'evoluzione tecnologica e l'avvento dei nuovi dispositivi, gli sviluppatori hanno sentito sempre di più la necessità di riuscire a rilevare l'effettivo comportamento degli utenti nell'uso d'interfacce utente e, di conseguenza, si sono attivati per identificare le azioni tipiche derivate da tale interazione.

La comparsa dei dispositivi mobili ha fatto sì che fossero definite nuove tipologie di azioni.

Il punto cruciale della rivoluzione apportata dai dispositivi mobili è stato sicuramente il passaggio da un'interazione basata sull'utilizzo di periferiche d'input (mouse e tastiera) ad un'interazione che potesse inserirsi nel nuovo contesto d'uso, intriso di portabilità e mobilità, che le periferiche mobili avevano introdotto. Il tipo d'interazione capace di reggere il peso di questo rivoluzionario cambiamento è stata, senza ombra di dubbio, l'interazione basata sul tocco

Gli sviluppatori hanno così definito un catalogo di azioni basate sul tocco (eventi touch) rilevabili durante l'utilizzo delle interfacce utente, facendole entrare di diritto nell'insieme delle azioni tipiche.

Le azioni sono state definite in gergo tecnico col nome di *touch gesture*. Sono azioni effettuate tramite un tipo d'interazione tipicamente compiuta per mezzo di una o più dita.

Per effettuare la raccolta dei dati relativi al comportamento dell'utente durante l'interazione con l'interfaccia utente, in MUSE abbiamo scelto di registrare l'intera sequenza di azioni prodotta dall'utente.

Questa scelta ha implicato un'attenta analisi, dopo i primi test in laboratorio, delle prime sequenze d'interazione ottenute. Da tali sequenze sono emerse diverse problematiche: la prima può essere identificata nel volume di dati ottenuto.

La problematica in questione ha creato i presupposti per una riflessione su quale, allo stato dell'arte, potesse essere la metodologia più corretta ed efficiente per effettuare un'acquisizione dati che rispettasse l'integrità dell'informazione rilevata.

Allo stesso tempo non volevamo che un volume di dati eccessivo comportasse cali di performance o stress eccessivo per il server dedicato all'elaborazione e alla memorizzazione dei dati.

Abbiamo quindi deciso di implementare la procedura di registrazione degli eventi attraverso cinque fasi:

- Rilevazione dei dati lato client (con l'uso delle librerie JQUERY ed HAMMER.JS).
- Memorizzazione temporanea con riordinamento temporale degli eventi rilevati e applicazione filtri.
- Compressione con algoritmo RLE dei dati raffinati.
- Pacchettizzazione dei dati compressi.
- Invio dei log al server per la memorizzazione su database (con l'uso della tecnica AJAX).

Da un punto di vista della qualità dei dati, perdere informazioni durante la registrazione o acquisire informazioni superflue poteva costituire un grande rischio per l'effettivo raggiungimento degli obiettivi del progetto e del lavoro dei valutatori.

Vista la continua evoluzione delle librerie in grado di compiere tale attività, il primo passo per la realizzazione di una valida rilevazione è stato il valutare in primis quanti e quali eventi fossero gestiti in WUP.

WUP, come anticipato nel capitolo precedente, permetteva la registrazione di tutti gli eventi rilevabili per mezzo delle procedure native del Document Object Model e della libreria JQUERY.

La prima decisione presa, dopo l'analisi appena descritta, è stata quella di non togliere al nuovo progetto la possibilità di rilevare queste azioni. Dunque sono state

integrate nuove librerie, a supporto di quelle già presenti che sono state aggiornate all'ultima versione utile.

Tuttavia, MUSE aveva la necessità di rilevare anche le touch gesture dei dispositivi mobili di ultima generazione. Per integrare questa funzionalità nell'applicazione è stata valutata una serie di librerie sviluppate in Javascript, tra cui la più completa e flessibile è risultata essere HAMMER.JS. Con l'utilizzo di questa libreria, per MUSE è stato possibile rilevare ogni tipo di touch gesture, comprese quelle definite *multi touch*, caratterizzate dall'interazione compiuta con più dita sullo schermo del dispositivo.

Dunque, le strade intraprese nell'identificazione degli eventi lato client sono risultate, in conclusione, tre:

- Eventi nativi del DOM
- Eventi rilevabili con JQUERY
- Eventi rilevabili con HAMMER.JS

La memorizzazione temporanea dei dati e il riordinamento temporale degli eventi sono stati pensati per attuare una raffinazione dell'informazione rilevata ed evitare incongruenze sia a livello di rilevazione che a livello di memorizzazione su database. La memorizzazione sul local storage di HTML5 ha permesso di salvare temporaneamente la sequenza di dati. Tale sequenza, durante l'interazione, viene via via popolata con gli eventi rilevati. Una volta ottenuto l'insieme dei dati, abbiamo ritenuto necessario far rispettare al sistema l'ordine temporale con cui le azioni si erano verificate. Tale approccio è stato applicato in seguito ad alcuni test effettuati in laboratorio, in cui era stato riscontrato un insieme di incoerenze nei dati memorizzati dal server sul database.

Per effettuare il riordinamento è stata utilizzata la proprietà *timestamp* come discriminante. Tale proprietà indica, sotto forma d'intero positivo, il tempo esatto in cui l'evento si è verificato.

Essendo il timestamp un numero intero positivo, abbiamo potuto compiere l'operazione di riordinamento degli eventi in maniera davvero semplice, utilizzando operazioni matematiche davvero poco complesse.

Oltre al riordino temporale degli eventi, i dati acquisiti necessitavano di un affinamento. Dopo alcuni test in laboratorio, MUSE creava sequenze di eventi

troppo popolate. Da una lettura approfondita dei log, abbiamo notato che le sequenze di eventi risultanti erano caratterizzate da una grandissima quantità di azioni legate al movimento.

Per esempio gli eventi di movimento *mousemove* e gli eventi *pinchin/pinchout* venivano prodotti consecutivamente per tutta la durata del movimento realizzato, generando sequenze di eventi dello stesso tipo che, a livello di visualizzazione, occupavano troppo spazio all'interno delle timeline rendendo meno leggibile e interpretabile la sequenza d'interazione. Uno scenario di questo tipo ha portato alla creazione di metodi per effettuare il filtraggio delle azioni relative al movimento. La strategia utilizzata dal filtro sviluppato per risolvere problematica è semplicemente quella di far registrare all'applicazione solo il primo e l'ultimo evento di queste sotto-sequenze, ricalcolando la distanza coperta dal movimento compiuto tramite la formula della distanza tra due punti (*Figura 10*) applicata alle coordinate cartesiane dei due eventi registrati.

$$d(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Figura 10: Formula della distanza tra due punti.

L'applicazione dei filtri sugli eventi legati al movimento ha comportato un incremento delle prestazioni di MUSE circa del 30%. Intervenendo in questo senso, è stato riscontrato un netto miglioramento della visualizzazione dei dati, visto il gran numero di eventi rimossi dalle timeline.

Da un punto di vista della qualità dei dati, MUSE ha certamente un insieme di dati da analizzare più raffinato rispetto a quello che WUP era solito gestire.

Basti pensare alla fase successiva, la compressione RLE.

Abbiamo deciso di effettuare una compressione dei dati per evitare che l'informazione acquisita fosse troppo pesante in termini di occupazione di memoria. Con lo scopo di non appesantire il server, per procedere nella compressione è stato scelto l'algoritmo Run-Length Encoding (RLE). RLE è un algoritmo ideato per una compressione di tipo *loseless* (senza perdita di dati). Agisce sulle sequenze di elementi identici, trasformando una lunga sequenza di oggetti dello stesso tipo in una sequenza più corta. La sequenza risultante, dopo l'avvenuta compressione, è formata

da un'istanza dell'oggetto tipo, un carattere speciale e il numero di occorrenze necessario a definire quante volte l'oggetto si sia ripetuto all'interno della sequenza originaria. Per esempio se avessimo una sequenza di otto tap, normalmente troveremmo la nostra sequenza scritta in questo modo:

Notazione standard = [tap, tap, tap, tap, tap, tap, tap, tap]

Con l'applicazione dell'algoritmo RLE la sequenza originaria dell'esempio descritto avrebbe un formato completamente differente, diventando la seguente:

Notazione RLE = tap*8

Analizzando la notazione appena descritta, è rilevante fare qualche considerazione sull'efficienza della compressione di quest'algoritmo. Siamo passati dall'occupare otto celle di memoria ad utilizzarne solo tre per esprimere la stessa informazione, originariamente fatta di otto tap consecutivi.

In questo semplice caso l'utilizzo della compressione RLE ha comportato un risparmio di spazio del 37 %. Pensando per un momento a quanto una compressione di questo tipo possa essere performante su grandi sequenze caratterizzate da molteplici ripetizioni di elementi identici, è possibile comprendere per quale motivo RLE sia stato scelto.

La fase successiva alla compressione dei dati è stata la pacchettizzazione di questi ultimi. Per alleggerire maggiormente la comunicazione client-server, nella realizzazione del logger è stata inserita una procedura che si occupa di spezzare le varie sequenze già compresse in RLE in più pacchetti, in maniera tale da inviare al server pacchetti di rete meno pesanti, rendendo la trasmissione dati più efficiente. Lato server l'applicazione controlla successivamente i dati con un'analisi pacchetto per pacchetto, riordinandoli attraverso un identificativo progressivo e ricomponendo le sequenze generando in seguito linee di log che saranno poi processate dal server per la successiva memorizzazione su database.

Quest'ultima procedura effettua delle richieste http verso risorse dell'applicazione attraverso cui possono essere attivate le funzionalità per inserire i log sul database. L'attivazione di queste funzionalità viene eseguita tramite la tecnica AJAX, che col

supporto della libreria JQUERY è stata semplificata grazie ad una sintassi meno complessa.

5.3.2 I nuovi dati rilevabili

Dopo aver spiegato il funzionamento della nuova funzionalità di acquisizione dati, sembra giusto analizzare e descrivere meglio le azioni che MUSE è in grado di rilevare, cercando di illustrare dettagliatamente l'ampia gamma di nuove informazioni che costituiscono un passo in avanti del programma rispetto al precedente WUP.

Come anticipato in precedenza, gli eventi rilevabili da MUSE provengono da tre approcci differenti:

- Eventi rilevabili con JQuery:

`click, dblclick, mousedown, mouseenter, mouseleave, mousemove, mouseover, mouseout, mouseup, scroll`

- Eventi rilevabili nativamente:

`Beforeunload, blur, change, closetaskpanel, focus, focusin, focusout, geolocationchange, keypress, keydown, keyup, load, onchange, opentaskpanel, orientationchange, pageview, resize, select, submit, unload`

- Eventi rilevabili con Hammer.js:

`singleTap, doubleTap, press, rotateStart, rotateEnd, rotateCancel, swipeLeft, swipeRight, swipeUp,`

`swipeDown, pan, panLeft, panRight, panUp, panDown, pinchStart, pinchEnd, pinchMove, pinchCancel, pinch`

Tra queste tre categorie di azioni, quella introdotta ex novo all'interno di MUSE è la categoria di eventi rilevabili grazie all'utilizzo di HaMMER.JS. L'utilizzo di questa libreria, ha permesso di rilevare non solo le touch gesture base, ma addirittura le azioni basate su un approccio multi-touch.

Inoltre, la grande potenzialità di HAMMER.JS di definire eventi composti si è dimostrata funzionale, con particolare riferimento alla possibilità di estendere anche eventi semantici ed eventi custom.

Più dettagliatamente, gli eventi che abbiamo deciso di gestire per mezzo della libreria HAMMER.JS hanno reso possibile l'ottenimento d'informazioni che gli altri due approcci non erano in grado di fornire, come per esempio la distanza, il coefficiente di scala negli eventi multi-touch, gli spostamenti su asse X ed asse Y, la velocità del movimento e la tipologia dei puntatori utilizzati.

5.3.3 I Bad Usability Smells

Con l'evoluzione delle tecnologie e delle tecniche di programmazione, nel corso degli ultimi anni gli sviluppatori web hanno cercato di definire nuovi approcci per riuscire a risolvere le problematiche legate all'usabilità delle applicazioni web. Una delle tecniche in questione è conosciuta col nome di *refactoring*. La tecnica di refactoring è una tecnica di programmazione adottata per ristrutturare parti di codice attraverso alterazioni strutturali interne, senza tuttavia alterare il comportamento esterno generato dal codice stesso. Ogni applicazione di refactoring opera su piccole parti di codice. Utilizzare questa tecnica in maniera sequenziale può produrre una significativa ristrutturazione. Vista la piccola dimensione dei tipici interventi di refactoring, applicando questa tecnica all'interno delle applicazioni risulta più difficile compromettere il funzionamento dell'intero sistema. In particolare, da un punto di vista di usabilità, nelle applicazioni web (come sottolineano Grigera et al in [3]) la tecnica di refactoring può essere utilizzata per risolvere problemi di usabilità senza alterare il funzionamento dell'intera applicazione. Le problematiche di

usabilità spesso possono essere difficili da individuare e prendono il nome di *Bad Usability Smell*.

Analizzando alcuni degli strumenti descritti nello stato dell'arte, come per esempio ClickTale ed CrazyEgg, si può notare come questi strumenti forniscano al valutatore di usabilità informazione statistica analizzando i dati dell'interazione effettuata da utenti reali. Tale approccio non è sembrato del tutto soddisfacente, visto che necessariamente tali dati per essere produttivi implicavano l'impiego e le competenze di esperti di usabilità.

L'approccio utilizzato in MUSE per la rilevazione dei Bad Usability Smell è stato realizzato dividendo concettualmente la procedura per raggiungere l'obiettivo in due fasi principali:

- Definizione dei singoli Bad Usability Smell
- Sviluppo dell'algoritmo di rilevazione

La definizione dei vari smell è stata creata per mezzo della tecnologia *XML Schema*. Gli XML Schema hanno infatti permesso di ricreare, in ambiente XML, le specifiche situazioni in cui gli smell si erano verificati. Dunque la definizione di uno smell è stata pensata come una sequenza di uno o più eventi (differenti o dello stesso tipo) a cui viene applicata, come discriminante, una soglia temporale. Calcolando le differenze tra il timestamp di un evento e quello di un evento immediatamente precedente è stato possibile effettuare un confronto tra tale differenza e la soglia impostata, in modo tale da disambiguare i casi appartenenti e non appartenenti allo smell di riferimento.

Da un punto di vista della rilevazione, è stato implementato un algoritmo specifico per identificare i bad usability smell.

La problematica che abbiamo dovuto affrontare, nella progettazione dell'algoritmo di rilevazione, è individuabile nel come riuscire a verificare, data una sequenza di elementi, la presenza di sequenze interne minori coerenti con una descrizione astratta fornita.

Questa problematica può considerarsi simile al pattern matching delle espressioni regolari. La differenza tra il nostro scenario e il pattern matching per mezzo delle espressioni regolari sta nel fatto che quest'ultime operino su un insieme finito di elementi. Il nostro algoritmo invece doveva riuscire nella rilevazione di sotto-

sequenze all'interno di una sequenza primaria avente lunghezza indefinita. Oltre a questo, avevamo anche la necessità di valutare la dimensione temporale degli elementi adiacenti.

L'algoritmo risultante è stato quindi sviluppato per applicare la seguente procedura:

1. Partizionamento dei log per tipologia d'evento
2. Selezione dei log relativi al tipo definito nella prima istruzione della definizione astratta (XML Schema)
3. Verifica del numero di ripetizioni dell'evento con il numero di ripetizioni definito nella descrizione astratta.
4. Dopo il passo tre abbiamo i log ordinati in maniera tale da aver ottenuto il primo elemento delle sequenze da cui partire, del tipo indicato dalla prima istruzione della descrizione astratta. Successivamente viene verificata la compatibilità con il numero di ripetizioni indicato nella descrizione astratta anche per gli eventi successivi. Il risultato di questo passaggio crea il primo "mattoncino" di ciascuna sequenza potenzialmente candidata ad essere un bad usability smell.
 - a. Per ogni altra regola della descrizione astratta l'algoritmo valuta se questa è relativa a un evento di tipo indefinito o finito. Se l'evento risulta indefinito all'interno delle regole, l'algoritmo memorizza le verifiche pendenti di tale istruzione e passa alla verifica successiva. Se invece l'evento è definito, la procedura seleziona i log relativi a quel tipo di evento, verificando per ciascuno di essi se è immediatamente successivo all'elemento di una sotto-sequenza candidata o, se non ci sono verifiche pendenti, se è successivo all'ultimo elemento di una sotto-sequenza candidata. Fatto questo passaggio l'algoritmo procede alla successiva valutazione dei criteri rimanenti espressi all'interno della regola di definizione.
 - b. Nel caso sia rilevato un log che soddisfi la valutazione di tutti i criteri, questo viene aggiunto in coda all'opportuna sotto-sequenza confermandola come potenziale candidata.
 - c. Le sotto-sequenze per cui non sono stati trovati log compatibili vengono quindi eliminate, in quanto ritenute superflue.

- d. Alla fine di tutte le iterazioni, le sotto-sequenze rimaste indicano la presenza dei bad-smell e permettono alla timeline di identificarle e visualizzarle.

5.4 I vantaggi di MUSE per il valutatore

Viste le premesse, è facile immaginare quanto il lavoro del valutatore di usabilità non sia semplice. L'analisi dell'usabilità è, come abbiamo visto nei capitoli introduttivi, un'attività complessa. Il suo risultato è strettamente legato alle potenzialità degli strumenti che il valutatore può utilizzare per compiere il proprio lavoro. Di conseguenza, va precisato che in precedenza strumenti in grado di effettuare analisi automatiche di questo tipo spesso sono stati in grado di fornire al valutatore un'informazione esclusivamente di tipo quantitativo.

La nostra mission, nella realizzazione di MUSE, è stata principalmente il rendere l'uso dello strumento molto più semplice al valutatore, cercando di cambiare anche la tipologia d'informazione per lui essenziale in fase di analisi, spostando i dati rilevati dall'ambito quantitativo a quello qualitativo. Tale cambiamento ha portato alla realizzazione di alcune funzionalità interattive di MUSE.

Tali funzionalità saranno descritte all'interno di questo capitolo, illustrando approfonditamente i criteri che hanno portato alla loro progettazione.

5.4.1 La nuova interfaccia responsive

Come anticipato nel capitolo 2, la comparsa dei dispositivi mobili ha apportato grandi cambiamenti nell'utilizzo delle applicazioni web. Il contesto d'uso è infatti diventato frenetico, quotidiano, portabile. Questo ha fatto sì che le applicazioni web utilizzate in mobilità apportassero grandi mutamenti anche nel comportamento dell'utente, il quale, potendo usare gli applicativi in ogni momento e in qualsiasi tipo di luogo, può essere considerato parte integrante di un contesto caratterizzato da più punti d'attenzione e da possibili e ripetute distrazioni.

Dunque è diventato rilevante per il successo delle applicazioni effettuare uno studio approfondito dell'usabilità, per eliminare le problematiche ad essa legate e favorire la nascita di soddisfazione negli utenti. Ma soprattutto è altrettanto importante che l'interfaccia sia ben studiata da un punto di vista della chiarezza dei contenuti presentati.

Per raggiungere un tale risultato, le interfacce utente hanno dovuto evolversi, adattandosi ai nuovi dispositivi e cercando di visualizzare i contenuti nel modo più congeniale al device in uso.

Tenuto conto di questo, affrontando le prime analisi dello scenario d'uso di MUSE, abbiamo pensato il valutatore come un fruitore generico di applicazioni web. Questo tipo di approccio ha portato noi sviluppatori a decidere in qual modo permettere a MUSE di essere utilizzato in un contesto mobile.

Allo stesso tempo ci siamo posti la domanda su quale strategia adottare nello sviluppo di un'interfaccia che potesse adattarsi ai dispositivi mobili senza un eccessivo costo in termini di tempo e risorse. Dunque ci siamo chiesti se la scelta di un'interfaccia dedicata per ogni tipo di dispositivo mobile ed ogni orientamento della periferica fosse il sentiero giusto da percorrere.

Per trovare soluzione ai nostri dubbi, abbiamo deciso di ridisegnare completamente l'interfaccia adottando la tecnica del *responsive design*, che abbiamo già ampiamente descritto nel capitolo due.

Il responsive design, come già anticipato, attua cambiamenti dinamici della disposizione degli elementi di un'interfaccia prendendo come parametro di riferimento la larghezza (in pixel) degli schermi delle periferiche su cui l'interfaccia stessa viene visualizzata. Tali cambiamenti dinamici sono definiti da regole presenti all'interno dei fogli di stile prodotti con il linguaggio CSS.

Per rendere l'interfaccia di MUSE adattabile a tutti i dispositivi mobili sono stati definiti differenti breakpoint, tenendo conto delle categorie di dispositivi più presenti sul mercato. Più precisamente, i breakpoint sono stati definiti classificando i dispositivi in quattro categorie principali ed hanno permesso la creazione di quattro regole CSS (*media-query*) che sono alla base dei cambiamenti dinamici dell'interfaccia dello strumento.

Categoria Dispositivo	Media Query
Piccoli schermi	@media (max-width: 767px)
Da piccoli a medi schermi	@media (min-width: 768px) and (max-width: 979px)
Da medi a grandi schermi	@media (min-width: 980px) and (max-width: 1199px) {
Solo grandi schermi	@media (min-width: 1200px)

Figura 11: Riepilogo delle media query di MUSE.

La tabella in *Figura 11* rappresenta i breakpoint applicati all'interfaccia di MUSE. Com'è possibile notare, le media-query del linguaggio CSS basano il loro funzionamento su due proprietà principali: *min-width* e *max-width*, rispettivamente indicanti la minima e la massima larghezza dello schermo ammessa dalla regola stessa.

Durante lo sviluppo di MUSE, l'interfaccia è stata testata con diverse periferiche. I dispositivi in questione sono stati scelti in base alle dimensioni degli schermi e al produttore del dispositivo. Un altro elemento che abbiamo considerato rilevante, nella scelta dei dispositivi, è stata la performance dei device stessi. È stata scelta una gamma di dispositivi di varia potenza, per poter effettuare i test dell'interfaccia anche su dispositivi più datati.

In particolare ci siamo serviti dei seguenti dispositivi mobili:

- *Samsung Galaxy S4*
Tipologia: Smartphone – Dimensione schermo: 5 pollici
- *Samsung Galaxy S3*
Tipologia: Smartphone – Dimensione schermo: 4.8 pollici
- *Samsung Galaxy S Advance*
Tipologia: Smartphone - Dimensione schermo: 4 pollici

- *Google Nexus 5*
Tipologia: Smartphone - Dimensione schermo: 4,95 pollici
- *Hamlet XZPAD 700*
Tipologia: Smartphone - Dimensione schermo: 7 pollici
- *Apple Ipad (prima versione)*
Tipologia: Smartphone - Dimensione schermo: 10 pollici

L'interfaccia di MUSE si divide in due tranches principali che sono correlate ai due ambiti d'uso che interessano l'applicazione:

- Interfaccia lato utente
- Interfaccia lato valutatore

L'interfaccia lato utente è quella parte dell'interfaccia contenente le viste relative alle procedure che l'utente intenzionato ad effettuare il test su una pagina web dovrà utilizzare.

Le parti principali di questa sezione dell'interfaccia sono due:

- Lista dei test di valutazione
- Pannello per inizio, fine e salto dei task

La lista dei test di valutazione lato utente è stata progettata utilizzando una tabella HTML5.

La tabella è stata realizzata, come si nota in *Figura 12*, con una struttura a quattro colonne, indicanti rispettivamente numero identificativo, nome e descrizione della valutazione. L'ultima colonna è dedicata al pulsante con cui l'utente è in grado di avviare il test scelto.

The screenshot shows a Firefox browser window with the address bar displaying 'MUSE - Available Evaluation...'. The page content includes the MUSE logo and a table with the following data:

#	Name	Description	Start
1	Airport Mobile Web Site Evaluation	Evaluation of Dallas Airport mobile Web site	Participate
2	Computer Site Pisa Website	Analysis of the web site of the computer science course of Pisa's University	Participate
3	Comune di Pisa	Comune di Pisa	Participate
4	Comune di Reggio Calabria	Comune di Reggio Calabria	Participate
5	Corriere della Sera	Quotidiano a tiratura nazionale	Participate
6	Edreams	Site di prenotazioni aeree	Participate
7	Enel	Azienda Energetica	Participate
8	English highways mobile	Evaluation of English Highways mobile website	Participate
9	Eni	Azienda Energetica	Participate
10	Evaluation of Autostrade.it - new version	Second evaluation of autostrade.it	Participate
11	Evaluation of the mobile application of Autostrade S.p.a	Evaluation test for the mobile version of Italian Autostrade Spa website	Participate

Figura 12: Lista valutazioni lato user.

Da un punto di vista dell'usabilità la visualizzazione della lista delle valutazioni lato utente è stata ben più performante su schermi di grandi dimensioni e su dispositivi mobili utilizzati con un orientamento landscape.

Durante i test ci siamo resi conto che con dispositivi mobili utilizzati con orientamento portrait, tale visualizzazione comportava l'estensione del contenuto ben oltre la larghezza massima degli schermi. Tale problematica è stata risolta con l'utilizzo delle regole CSS, presenti in Twitter Bootstrap, per la creazione di tabelle responsive.

L'approccio adottato dal framework di Twitter per la creazione di timeline adattive si basa sull'utilizzo della proprietà *overflow* dei fogli di stile CSS.

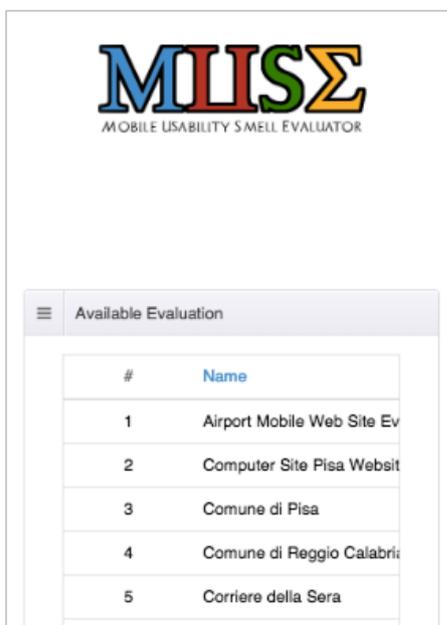
La proprietà *overflow* definisce come il browser debba comportarsi nel visualizzare contenuti che oltrepassino, per dimensioni, la larghezza o l'altezza (o entrambe) dell'elemento in cui sono contenuti.

La proprietà `overflow` può assumere sei possibili valori:

- *Visible*:
Il contenuto non viene troncato. La parte eccedente viene visualizzata al di fuori dell'elemento contenitore. Il valore `visible` è il valore di default nel caso la proprietà `overflow` non venga dichiarata o assegnata.
- *Hidden*:
Il contenuto in eccesso viene troncato e la parte in eccesso è resa invisibile.
- *Scroll*:
Il contenuto viene troncato e la parte in eccesso viene visualizzata per mezzo di barre di scorrimento, verticali e/o orizzontali a seconda dello scenario. È possibile decidere manualmente quale barra di scorrimento attivare attraverso i valori `scroll-x` e `scroll-y`.
- *Auto*:
Molto simile a `scroll`, con questa impostazione il contenuto viene troncato e la parte in eccesso viene visualizzata per mezzo di barre di scorrimento create automaticamente dal browser.
- *Initial*:
Imposta la proprietà `overflow` al suo valore predefinito.
- *Inherit*:
Eredita il valore della proprietà dall'elemento contenitore. Per il funzionamento del valore `inherit` è necessario che l'elemento contenitore abbia una proprietà `overflow` settata su uno dei valori ammessi dalla proprietà.

Più precisamente, per gestire tabelle HTML con contenuti più grandi delle dimensioni dello schermo, Twitter Bootstrap fa uso della proprietà `overflow` settata su `hidden`, per quanto riguarda l'asse Y, e su `auto`, per l'asse X.

Tale approccio permette di visualizzare il contenuto in eccesso, nascosto in senso orizzontale, per mezzo di barre di scorrimento. Allo stesso tempo lascia invariato il comportamento del browser da un punto di vista della verticalità, facendo sì che l'attività di scroll verticale rimanga legata allo scorrimento dell'intera pagina.



#	Name
1	Airport Mobile Web Site Ev
2	Computer Site Pisa Websit
3	Comune di Pisa
4	Comune di Reggio Calabria
5	Corriere della Sera

Figura 13: Visualizzazione della lista valutazioni di MUSE con layout portrait.

Oltre a questo elemento, il rifacimento dell'interfaccia lato utente ha coinvolto anche l'elemento che in WUP era conosciuto come *task panel*, un pannello attraverso il quale si rendeva possibile iniziare e/o concludere il task oppure, in alternativa, saltarlo per passare al task successivo.

Gli interventi effettuati sul task panel sono risultati necessari per l'applicazione dell'approccio responsive al pannello stesso, rendendolo così fruibile e correttamente visualizzabile su ogni tipo di dispositivo. In aggiunta sono stati apportati cambiamenti cromatici, per avere una linea di colori coerente con il resto degli elementi dell'interfaccia. La modifica non è stata applicata solo e soltanto all'elemento contenitore, ma anzi è stata introdotta su tutti gli elementi interni del task panel, rendendo adattiva sia l'informazione testuale contenuta nel pannello che gli stessi pulsanti presenti.

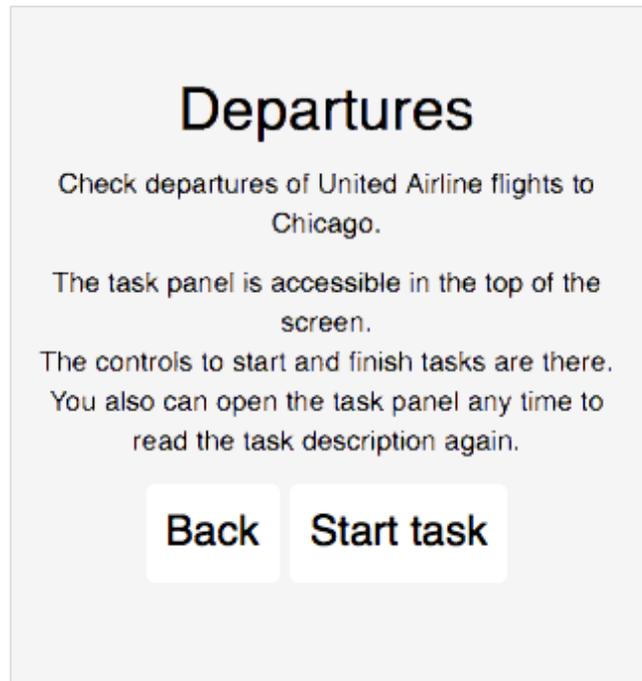


Figura 14: Esempio tipico del task panel di MUSE.

Com'è possibile notare in *Figura 14*, il task panel contiene i comandi per tornare alla lista delle valutazioni e per iniziare il task. È necessario precisare che i pulsanti presenti nel task panel sono stati progettati in modo tale da attuare cambiamenti dinamici al proprio contenuto in ragione dello scenario prodotto dall'utente. Più precisamente, quando l'utente non ha ancora iniziato a compiere il task, i comandi disponibili sono solo quelli attivabili dal tasto *back* e *start task*.

Nel caso in cui invece l'utente abbia fatto click o tap sul pulsante *start task* e quindi avviato il task, i pulsanti visualizzati all'interno del pannello diventeranno *back*, *finish task* e *skip task*.

Va sottolineato che il tasto *skip task* è visualizzabile solo ed esclusivamente se il task in questione è stato configurato dal valutatore come uno di quelli abilitati all'essere saltato.

Dopo aver analizzato i cambiamenti apportati all'interfaccia lato utente, sembra opportuno elencare quali siano le modifiche alla visualizzazione dedicate a rappresentare le varie procedure necessarie al valutatore per un corretto utilizzo dello strumento.

Questa parte della realizzazione del progetto ha comportato una grande quantità di lavoro che può essere sintetizzato in tre momenti principali:

- Adeguamento della visualizzazione delle vecchie viste utilizzate in WUP ed ereditate da MUSE.
- Creazione di timeline interattive per la visualizzazione delle sequenze di eventi.
- Creazione della sezione *overview*, dedicata alla visualizzazione di una panoramica dettagliata degli elementi e dell'andamento dei singoli task.

Per ciò che concerne il primo punto dell'elenco sopra descritto, il primo passo per la realizzazione dell'interfaccia lato valutatore è stato quello di identificare tutte le funzionalità dedicate all'inserimento, modifica, cancellazione e configurazione delle componenti fondamentali delle valutazioni, con particolare riferimento a valutazioni, task, custom semantic event e log.

Abbiamo riprogettato questa parte dell'interfaccia realizzando viste analoghe a quelle già esistenti ed aggiungendo a queste la capacità di adattamento agli schermi dei dispositivi mobili. Questa scelta è stata presa per due motivazioni essenziali. La prima può essere riscontrata nella coerenza, dato che non ci sembrava affatto vantaggioso per il valutatore avere una visualizzazione completamente stravolta rispetto a quella con cui era abituato ad interagire. La seconda motivazione invece può essere identificata nella volontà di mantenere invariato il paradigma d'interazione utilizzato in precedenza, anche durante l'utilizzo dello strumento per mezzo di device mobili.

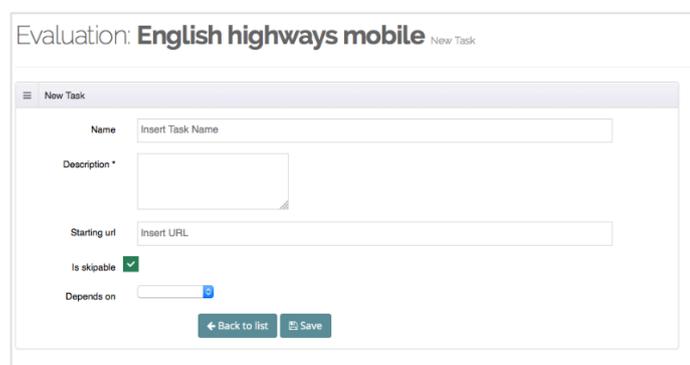


Figura 15: Esempio di riprogettazione di una precedente visualizzazione.

Gli altri due punti elencati nell'elenco precedente possono essere considerati la parte più interessante di MUSE da un punto di vista della visualizzazione. Sono questi gli argomenti che costituiscono il focus di questa tesi. Per una comprensione dettagliata circa ciò che concerne la realizzazione delle nuove timeline interattive e della sezione overview si rimanda rispettivamente ai paragrafi 5.4.3 e 5.4.5 di questo capitolo.

5.4.2 *L'acquisizione degli screen-dump*

Una parte fondamentale per lo sviluppo della visualizzazione interattiva di MUSE è costituita dai cosiddetti screen-dump delle pagine web analizzate. Gli screen-dump sono immagini raffiguranti l'intera pagina web su cui si è verificata l'interazione degli utenti e possono essere considerati uno degli strumenti più utili per i valutatori di usabilità in quanto aiutano gli stessi a comprendere meglio come le pagine oggetto di test siano state presentate all'utente.

MUSE fa uso di queste immagini, in particolare per la visualizzazione degli eventi *pageview* e dei problemi di usabilità rilevabili tramite degli indicatori di cattiva usabilità conosciuti col nome di Bad Usability Smell.

D'importanza fondamentale per l'acquisizione di queste immagini all'interno di MUSE sono stati l'approccio proxy-based utilizzato dallo strumento, i web worker di HTML5 e la libreria HTML2CANVAS. Queste tre componenti costituiscono il fondamento della procedura di acquisizione, rendering e memorizzazione delle immagini in questione.

La procedura di acquisizione degli screen-dump ha il suo inizio con la rilevazione dell'evento *pageview*, un evento semantico inserito, a livello di sistema, nello strumento. Attraverso tale evento MUSE è in grado di rilevare quando tutte le risorse necessarie alla generazione della pagina web sono state caricate. La rilevazione di questo evento implica, da un punto di vista d'interazione, un caricamento iniziale di pagina o, in alternativa, la volontà dell'utente di compiere un salto di navigazione verso una pagina web differente da quella con cui stava interagendo. Fatte queste considerazioni, abbiamo ritenuto coerente avviare l'acquisizione della funzionalità di screen-dump ogni qual volta un evento *pageview* fosse rilevato.

Più precisamente, la procedura d'acquisizione delle immagini è stata sviluppata attraverso i seguenti passi:

1. Rilevazione evento pageview.
2. HTML2CANVAS viene lanciato, una volta caricata la pagina, e genera attraverso il rendering effettuato con l'elemento canvas di HTML5 una stringa definita col nome di *data-url*. Tale stringa impartisce al browser istruzioni specifiche su come ridisegnare l'immagine raffigurante la pagina web.
3. Pacchettizzazione dell'immagine e invio.
4. Nel caso in cui l'utente effettui un salto di pagina o si debba inviare l'immagine al server, viene attivato un web worker delle HTML5 API. I web worker sono procedure in grado operare in background rispetto agli script in esecuzione sulla pagina principale. L'utilizzo dei web worker ha reso possibile l'acquisizione d'immagini anche durante i salti di navigazione, dato che attraverso questo tipo di funzionalità abbiamo potuto gestire eventuali situazioni in cui si rendeva necessaria una re-inizializzazione della procedura, tenendo conto dell'ultimo pacchetto dati inviato. Quest'approccio ha permesso di far ripartire l'acquisizione dal punto in cui si era fermata, evitando così di ottenere immagini troncate o addirittura corrotte.
 - a. I web worker utilizzati, per effettuare il loro compito, sono stati inizializzati con delle informazioni necessarie alla corretta gestione delle problematiche appena descritte:
 - Timestamp (che dà il nome al web worker)
 - Numero del pacchetto
 - Ticket Id
 - Nome del file da memorizzare sul server
 - Dimensione originaria dell'immagine
 - Dominio di riferimento della pagina da acquisire
 - URL della pagina

Quanto alle problematiche riscontrate in fase di sviluppo, la principale difficoltà riscontrata nell'acquisire le immagini rappresentanti le pagine web è stata riscontrata in una limitazione della libreria Html2canvas.

HTMLTOCANVAS, nella sua implementazione di base, è in grado di acquisire correttamente solo immagini di pagine web in cui la libreria è direttamente inclusa.

In fase di sviluppo siamo riusciti ad aggirare tale limitazione proprio grazie all'utilizzo del server proxy. Quest'approccio ci ha permesso di prendere completamente il codice di pagine esterne e di rieffettuare il corretto rendering dei fogli di stile internamente all'applicazione. Inserendo la libreria HTML2CANVAS all'interno delle pagine ricreate dal proxy abbiamo potuto acquisire l'immagine in maniera corretta, producendo screen-dump fedeli alle pagine originarie, anche se originariamente esterne al dominio di riferimento dell'applicazione.

5.4.3 Le nuove timeline

Compiere la valutazione dell'usabilità su applicazioni web implica l'acquisizione, per lo più dettagliata possibile, del comportamento dell'utente in seguito all'interazione dello stesso sull'interfaccia utente dell'applicazione utilizzata. Avere una sostanziosa quantità di dati circa le modalità con cui l'utente ha interagito con l'interfaccia non è tuttavia, ai fini di una corretta valutazione, l'unica componente utile per il valutatore.

Per permettere ai valutatori di raggiungere tale obiettivo è indispensabile progettare e realizzare una visualizzazione in grado di rendere significativa la mole di dati acquisiti, facilitando l'analisi e l'interpretazione dell'informazione.

Come già anticipato nello stato dell'arte, in campo di usabilità gli approcci utilizzati dagli strumenti descritti per la visualizzazione dei dati sono svariati. Questi permettono di valutare gli strumenti per la valutazione di usabilità anche da un punto di vista della rappresentazione dei dati stessi, dando al valutatore la possibilità di scegliere quale tra questi strumenti sia quello più conveniente per il tipo di valutazione o analisi che deve compiere.

Prima di analizzare nel dettaglio le modalità di visualizzazione dei dati fornite dal nostro strumento, sembra corretto e coerente precisare quali siano stati gli obiettivi e

le esigenze a cui MUSE avrebbe dovuto rispondere, motivando le scelte fatte in base ad un'attenta osservazione della visualizzazione utilizzata da altri strumenti.

L'obiettivo fondamentale di MUSE è individuabile nella volontà di ottenere informazioni inerenti il comportamento dell'utente. Tali dati, per il raggiungimento dell'obiettivo, dovevano non solo procurare informazioni utili sull'interazione, ma anche informazioni aggiuntive ai fini di una rilevazione del contesto d'uso dei dispositivi.

Dunque la valutazione dell'usabilità effettuata con MUSE è stata pensata proprio in questo senso, cercando di renderla una valutazione standard dell'usabilità ma allo stesso tempo anche una valutazione contestuale.

Con la realizzazione di MUSE si è cercato di far evolvere la stessa valutazione dell'usabilità ad un livello più alto, introducendo nuove funzionalità nello strumento senza eliminare le potenzialità introdotte da WUP, il suo predecessore.

Un altro aspetto che abbiamo analizzato attentamente durante la fase di sviluppo è stata l'operazione di confronto tra le sessioni di interazione registrate. Molti degli strumenti citati nello stato dell'arte generano una rappresentazione visuale dell'interazione per mezzo di mappe di calore o, addirittura, registrando video con cui l'intera sequenza d'interazione degli utenti può essere visionata a posteriori.

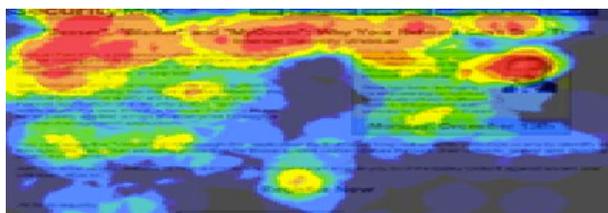


Figura 16: Esempio di mappa di calore.

Applicare tali tipi di visualizzazione per un eventuale confronto tra interazioni di due o più utenti diversi comportava non poche difficoltà nella successiva interpretazione dei dati. Essendo visualizzazioni indipendenti, realizzare confronti visuali con questi strumenti comportava al valutatore il limite di dover analizzare le immagini una per volta, senza avere la possibilità di ottenere una visione chiara di eventuali problemi di usabilità.

Viste queste considerazioni, per la visualizzazione dell'interazione sulle interfacce sono state realizzate timeline temporali interattive.

La visualizzazione per mezzo di timeline interattive è stata ritenuta, in fase progettuale, la più idonea per realizzare rappresentazioni visuali dell'intera sequenza di azioni compiute dall'utente, garantendo maggiore flessibilità nello sviluppo delle funzionalità dedicate al confronto tra le varie sessioni effettuate.

Il primo passo compiuto per realizzare le timeline interattive è stato l'identificare una libreria che garantisse, con un riferimento temporale ben delineato, una visualizzazione chiara delle sequenze di eventi. Allo stesso tempo volevamo una libreria che fosse, a livello di visualizzazione, coerente con i criteri che hanno portato allo sviluppo e alla rivisitazione di tutti gli altri elementi dell'interfaccia. Dunque la realizzazione delle timeline interattive è stata compiuta con la già citata libreria Almende CHAP LINKS LIBRARY, sia per la chiarezza con cui è in grado di visualizzare gli eventi sia per l'uso del responsive design che ha permesso alle timeline di adattarsi ad ogni tipo di schermo, indipendentemente dall'orientamento.

Effettuando i primi test sulla libreria per arrivare alla creazione del modello di timeline da utilizzare, l'analisi dei primi risultati ha portato la necessità di definire a priori quali tipi di informazione tali timeline dovessero contenere. Le informazioni da noi selezionate, ritenute da noi fondamentali per la valutazione, hanno comportato un'attenta stima degli spazi da utilizzare, cercando di mantenere coerenza, chiarezza e precisione nella visualizzazione.

Più precisamente, le informazioni in questione sono state individuate in:

- Identificativo della sessione
- Informazioni sul browser
- Informazioni sul dispositivo
- Data della sessione
- Tempo totale della sessione
- Sequenza degli eventi

Per visualizzare i dati elencati è stato deciso di inserire le timeline all'interno di elementi contenitori, realizzati con un approccio responsive, la cui testata è stata utilizzata per la rappresentazione dei dati rappresentanti i primi cinque punti dell'elenco sopra descritto.



Figura 17: Esempio di testata del pannello contenente la timeline.

Come si può notare in *Figura 18*, il numero identificativo della sessione, e di conseguenza della timeline, è stato posizionato sulla sinistra della testata del pannello.



Figura 18: Visualizzazione del numero identificativo delle timeline.

Sulla parte più destra della testata è stata realizzata la visualizzazione delle informazioni relative a browser, sistema operativo e device utilizzati dall'utente, con l'aggiunta a lato della data in cui la registrazione è avvenuta e del tempo impiegato per compiere il task di riferimento.

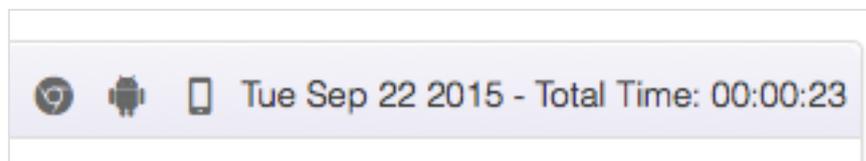


Figura 19: Visualizzazione delle informazioni su browser, sistema operativo, data e tempo impiegato.

La parte di testata appena descritta (il lato destro) è una delle parti interattive della timeline.

Come si nota in *Figura 20*, le informazioni relative a browser, sistema operativo e modello di dispositivo utilizzato sono rappresentate da icone.

Queste sono state realizzate utilizzando FONT AWESOME (ver. 4.4.0), un font vettoriale appositamente rilasciato per lo sviluppo di pagine web realizzate con approccio responsive.

Le icone realizzate sono state rese interattive. L'interazione di questi elementi è stata gestita per mezzo dell'evento *hover* nel caso in cui il valutatore utilizzi un

dispositivo desktop per visualizzare le timeline. L'interazione per mezzo di dispositivo mobile invece è stata gestita con l'evento *tap*.

Più precisamente, com'è possibile notare in *Figura 20*, quando il valutatore posiziona il puntatore del mouse o effettua un tap su una di queste icone, sotto la stessa diventa visibile una piccola didascalia (tecnicamente definita col nome di *tooltip*) che riporta in forma più dettagliata i dati relativi (browser, sistema operativo e device).

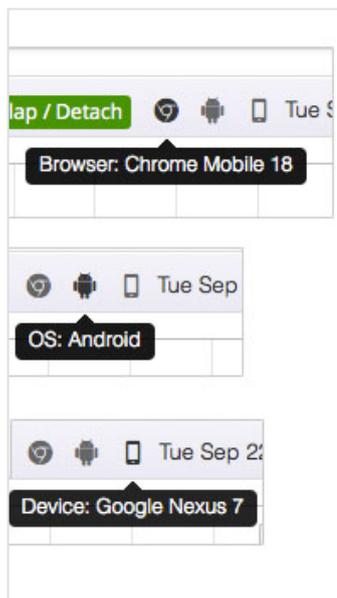


Figura 20: Esempio di click/hover sulle icone.

Passando alla visualizzazione dell'intera sequenza d'interazione compiuta dagli utenti, questa viene rappresentata attraverso la visualizzazione creata con la libreria Almende CHAPS LINKS LIBRARY. Come già detto in precedenza, la visualizzazione grafica delle sequenze di eventi creata da questa libreria è in grado di creare timeline interattive capaci di adattare il proprio stile alla larghezza del display del dispositivo utilizzato dagli utenti.

In *Figura 21* è possibile notare quale sia stata la rappresentazione scelta per le sequenze di eventi all'interno di MUSE:

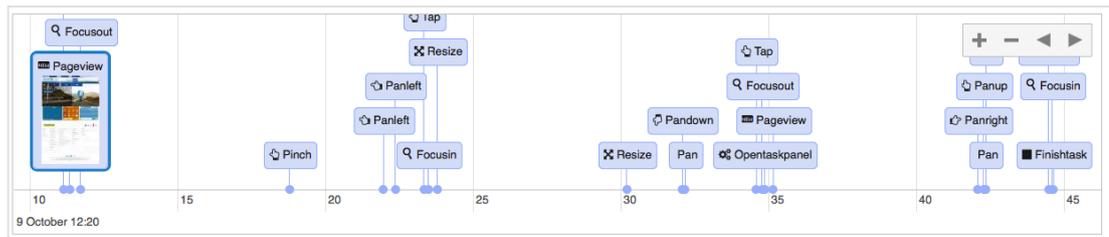


Figura 21: Esempio di visualizzazione delle timeline di MUSE.

Come si nota dalla precedente immagine, la timeline interattiva è composta, nella parte dedicata alla visualizzazione delle sequenze, dei seguenti elementi:

- Box contenitori interattivi, rappresentanti ogni azione registrata.
- Asse delle ascisse, posizionato sotto i box interattivi, indicante la scala temporale di riferimento
- Controlli interattivi per le funzionalità di manipolazione delle timeline (zoom e spostamento)

Prima di introdurre la veste grafica e le varie funzionalità delle timeline interattive, si ritiene utile fare una precisazione per ciò che concerne le modalità di attivazione e disattivazione dell'interattività delle timeline stesse.

Per evitare possibili errori da parte dei valutatori, è stato scelto un approccio in base al quale le timeline permettono l'interazione solo dopo aver fatto un click od un tap (se si utilizza un dispositivo mobile) su di esse.

Capire se una timeline è stata attivata o no risulta davvero molto semplice, dato che a livello visuale la timeline attivata risulta caratterizzata da un bordo blu che segnala la possibilità di interagire con la stessa. (vedi Figura 22).

Dunque la timeline, quando non attivata, non è caratterizzata da tale bordo. Per disattivare la timeline sarà necessaria solo ed esclusivamente l'interazione dell'utente con un qualunque elemento dell'interfaccia esterno alla timeline stessa.

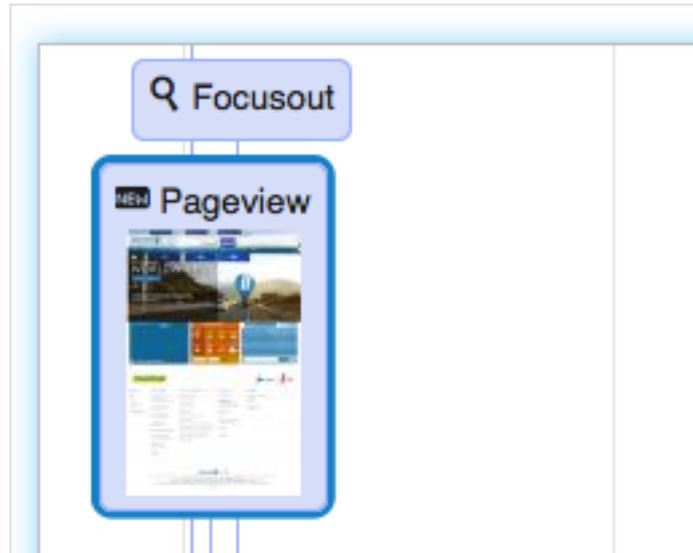


Figura 22: Esempio di timeline attivata.

I box utilizzati per la visualizzazione degli eventi sono stati realizzati per mezzo di elementi HTML5. Ciò ha permesso di rendere tali box interattivi, consentendo l'attivazione di funzionalità specifiche per alcune tipologie particolari di evento.

A supporto delle funzionalità aggiuntive sono stati utilizzati i cosiddetti *data-attributes* (*data-**) di HTML5. In particolare, per ogni evento rappresentato della timeline sono stati definiti tali attributi indicanti le informazioni relative all'evento rilevato.

Più precisamente per gli eventi generici sono state definite, sotto forma di *data-attribute*, le coordinate *x* ed *y* indicanti il punto in cui, in un riferimento cartesiano, è stato rilevato l'evento stesso.

Da un punto di vista dell'interazione gli eventi generici possono essere oggetto di eventi *click* o *tap*. Al verificarsi di questi due eventi sui box, questi si evidenziano assumendo come sfondo del box il colore giallo e applicando al box stesso un bordo leggero color arancio (vedi *Figura 23*).

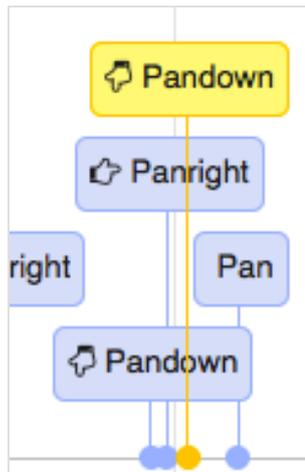


Figura 23: Esempio di evidenziazione di un generico evento.

Per aumentare la chiarezza dell'informazione all'interno dei box indicanti gli eventi, abbiamo deciso di inserire delle icone accanto alle etichette delle azioni rilevate.

Tali icone sono state utilizzate per favorire un riconoscimento degli eventi attraverso l'informazione semantica delle icone stesse.

Tale approccio si è rivelato funzionale soprattutto negli eventi che coinvolgevano attività di movimento.

Come si può notare in *Figura 23*, per tali eventi sono state utilizzate icone raffiguranti frecce o mani (nel caso di eventi mobile) nella stessa direzione intrapresa nel movimento caratterizzante l'azione compiuta.

Da un punto di vista visuale, di particolare interesse sono gli eventi definiti col nome di *pageview*.

Pageview è un evento semantico scatenato dall'interfaccia all'avvenuto caricamento di tutte le risorse necessarie alla generazione della pagina web visualizzata.

Tale evento indica al valutatore anche un plausibile salto di navigazione da parte dell'utente. La rilevazione di pageview, come già anticipato precedentemente, costituisce il discriminante per l'eventuale avvio della procedura di acquisizione degli screen-dump.

Per questa ragione abbiamo deciso di inserire all'interno dei box di pageview, una miniatura dello screen-dump generato dopo la rilevazione (vedi *Figura 24*).

L'informazione riguardante l'immagine generata è stata descritta all'interno degli stessi box per mezzo degli attributi *data-** di HTML5.

Tali attributi, nel caso di pageview, sono stati definiti in maniera diversa rispetto a quelli implementati all'interno dei box degli eventi generici.

Più precisamente sono stati aggiunti, per l'evento semantico pageview, i seguenti tre attributi:

- *data-screenshot:*
attributo custom indicante la posizione del file dello screenshot all'interno del server.
- *data-original-width:*
attributo custom che indica la larghezza originaria dello schermo del dispositivo da cui è stato generato lo screenshot.
- *data-original-height:*
attributo custom che indica l'altezza originaria dello schermo del dispositivo da cui è stato generato lo screenshot.

Da un punto di vista dell'interattività, i box degli eventi pageview permettono di visualizzare, con le dimensioni originarie, lo screenshot della pagina web a cui si riferiscono.

Tale funzionalità è stata realizzata per mezzo delle finestre di dialogo di Twitter Bootstrap ed è attivabile effettuando un doppio click o un doppio tap (in caso di dispositivo mobile) sulla miniatura presente all'interno del box.



Figura 24: Esempio di box interattivo per l'evento pageview.

Un altro elemento di grande rilevanza nelle timeline interattive di MUSE è il pannello contenente i controlli dedicati alle funzionalità di manipolazione e scorrimento delle timeline stesse (vedi *Figura 25*).

Tale pannello è accanto al vertice destro della timeline ed è composto da quattro pulsanti principali:

- pulsante per la funzionalità zoom in
- pulsante per la funzionalità zoom out
- pulsante per lo scorrimento verso sinistra
- pulsante scorrimento verso destra

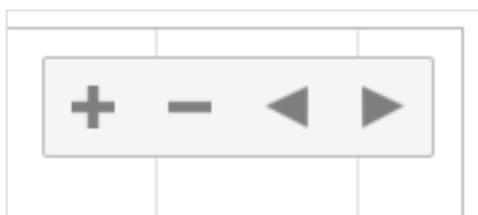


Figura 25: Pannello per zoom e navigazione della timeline.

Le funzionalità di *zoom in* e *zoom out* effettuano un cambiamento dinamico della visualizzazione della timeline. Più precisamente queste due feature cambiano la scala temporale delle timeline, permettendo al valutatore di trasformare la visualizzazione degli eventi e renderla, nel caso gli eventi siano molti, più leggibile.

Le scale temporali ammesse dalla libreria Almende CHAP LINKS LIBRARY sono racchiuse in un intervallo che va dal millisecondo fino ad arrivare all'anno solare.

Tale scala può essere impostata manualmente ma, per un corretto funzionamento delle funzionalità di zoom, abbiamo deciso di utilizzare l'impostazione *auto*.

Con tale impostazione, la scala temporale cambia dinamicamente in base alla quantità di zoom richiesta dal valutatore. I pulsanti per gestire la funzionalità di zoom sono individuabili tramite il segno + e il segno -.

Per ciò che concerne invece i comandi di navigazione delle timeline, i due pulsanti rimanenti (le due frecce) consentono al valutatore di poter scorrere la timeline orizzontalmente, in entrambe le direzioni, permettendo al valutatore di navigare la stessa e recarsi in un punto ben preciso della sessione visualizzata. Va precisato che tale funzionalità è attivabile anche mediante un'interazione basata su periferiche di input e tocco.

A timeline attivata, tenendo cliccato il mouse, o posizionando una delle dita sulla timeline, è possibile infatti muovere la timeline a proprio piacimento. L'attivazione della navigazione è gestita, per l'interazione col tocco, dall'evento *press*.

Altre due funzionalità di grande rilevanza che abbiamo introdotto in MUSE sono procedure dedicate al filtraggio degli eventi della sequenza visualizzata.

Per rendere tali procedure interattive, abbiamo realizzato un pannello attraverso il quale è possibile attivare la funzionalità appena citata (vedi *Figura 26*).

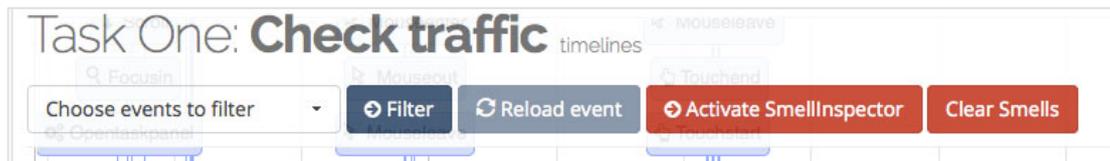


Figura 26: Pannello per funzionalità avanzate (Bad Smell e filtri).

Sul pannello delle funzionalità avanzate sono stati posti i comandi per l'attivazione di tali strumenti.

Più precisamente, analizzando il pannello nel dettaglio possiamo trovare i seguenti elementi:

- Menu a tendina per la scelta multipla degli eventi da filtrare all'interno delle timeline.
- Tasto "*Filter*":
Pulsante attraverso cui è possibile attivare il filtraggio degli eventi.
- Tasto "*Reload event*":
Pulsante con cui è possibile ripristinare la visualizzazione originaria della timeline dopo un avvenuto filtraggio degli eventi.

- Tasto “*Activate Smell Inspector*”:
Pulsante mediante il quale è possibile attivare l’identificazione e la visualizzazione dei Bad Usability Smell sulla sequenza di eventi rappresentata all’interno della timeline.
- Tasto “*Clear Smells*”:
Pulsante con cui è possibile riportare la timeline all’aspetto originario dopo l’eventuale attivazione dell’identificazione e rappresentazione dei Bad Usability Smell.

Descrivendo nel dettaglio l’interazione fornita dagli elementi appena descritti, le funzionalità di filtraggio degli eventi sono state realizzate per mezzo di menu a tendina a scelta multipla. Tali menu sono stati implementati utilizzando un elemento *select* di HTML5, fornito dal framework Twitter Bootstrap, avente la caratteristica peculiare di rendere possibile la selezione multipla delle opzioni di scelta messe a disposizione dal sistema.

Quando l’elemento *select* viene azionato, la tendina corrispondente rimane aperta consentendo al valutatore di scegliere uno o più tipi di eventi da nascondere all’interno della timeline.

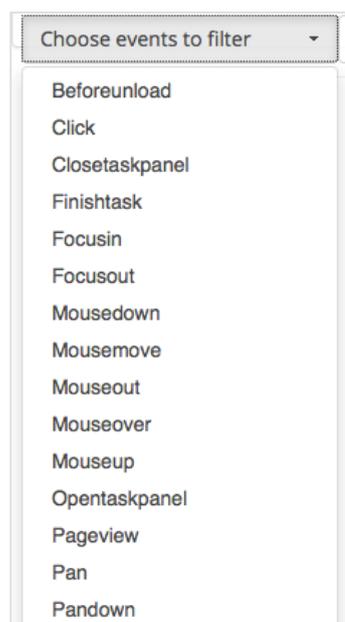


Figura 27. Select a scelta multipla per effettuare il filtraggio degli eventi.

Una volta effettuata una o più selezioni, il valutatore può attivare il filtraggio degli eventi cliccando il tasto *Filter* posizionato accanto al menu a tendina.

Con l'attivazione della funzionalità di filtraggio i box degli eventi appartenenti alla tipologia o alle tipologie scelte vengono nascosti ed eliminati temporaneamente dalla sequenza di eventi della timeline.

L'occultamento dei box è stato implementato per mezzo di JQUERY, che modifica lo stile CSS dei box alterando la proprietà *display*, che viene impostata su *none*.

Allo stesso tempo, è stato necessario memorizzare temporaneamente gli eventi nascosti, per l'eventuale necessità del valutatore di renderli di nuovo visibili all'interno della timeline.

Interagendo con il tasto *filter*, l'algoritmo sviluppato memorizza il numero identificativo degli eventi da nascondere all'interno di un array Javascript.

Nel caso in cui il valutatore decida di ripristinare gli eventi nascosti all'interno delle timeline, attraverso l'interazione con il tasto *Reload event* l'algoritmo dedicato al ripristino effettua una lettura di tutti gli elementi presenti nell'array contenente gli identificativi degli eventi occultati.

Con un'iterazione effettuata elemento per elemento, l'algoritmo interpreta il numero identificativo dell'evento e cerca all'interno del Document Object Model un qualunque box, con proprietà *display* settata su *none*, avente l'attributo *data-dbid* impostato con un valore identico a quello appena letto dall'array. Ad avvenuta verifica della corrispondenza di tali valori, al box in questione è applicata la procedura inversa al filtraggio, reimpostando la proprietà *display* al valore originario e attribuendole quindi il valore *block*.

Giusto per fare un esempio dell'attività di filtraggio, com'è possibile notare in *Figura 28 e 29*, i box degli eventi panleft scelti per l'esempio sono stati nascosti dalla timeline.

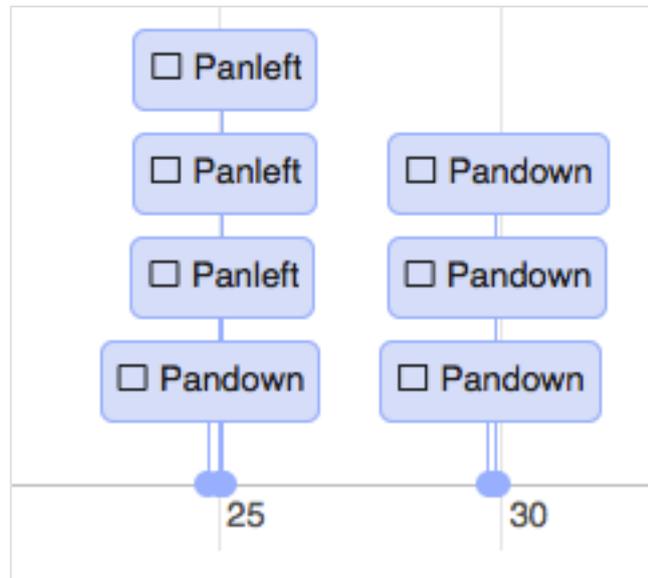


Figura 28. Esempio di eventi panleft prima del filtraggio.

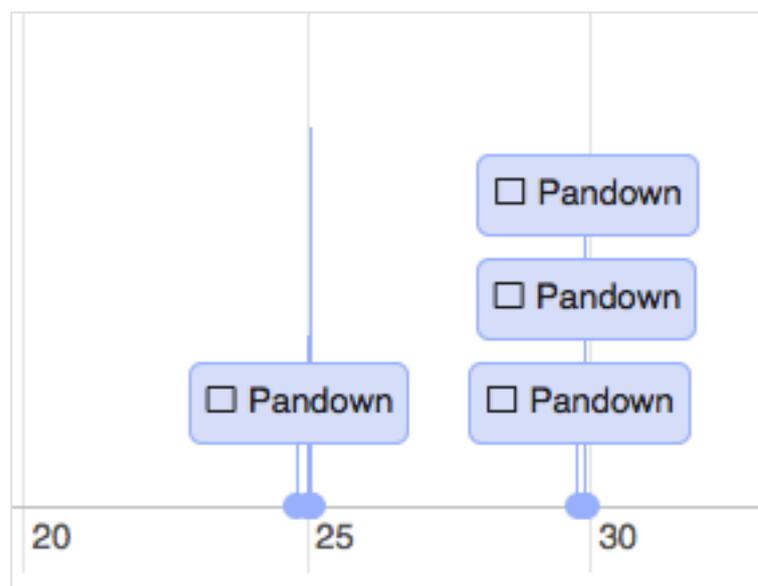


Figura 29. Esempio di eventi panleft nascosti dal filtraggio.

L'altro strumento avanzato che è possibile attivare dal pannello è, come descritto nell'elenco, lo *Smell Inspector*. L'algoritmo dedicato alla rilevazione dei bad usability smell viene attivato interagendo con il tasto "Activate Smell Inspector".

Durante la fase di sviluppo di questa funzionalità ci siamo chiesti quali fossero, nelle fasi preliminari all'identificazione, le possibili necessità del valutatore in merito a tale compito. Abbiamo individuato, in primis, la necessità per il valutatore di poter scegliere a priori, prima di lanciare la procedura di rilevazione, quale dei vari smell

rilevare. Inoltre ci è sembrato coerente permettere allo stesso di scegliere su quali sessioni di interazione applicare la rilevazione dei bad usability smell .

Per trovare soluzione a tali bisogni, abbiamo deciso di implementare un form HTML dedicato alla scelta delle tipologie di bad usability smell da rilevare e delle timeline su cui attivare l'identificazione degli smell stessi.

All'interno del form sono stati utilizzati degli operatori di scelta multipla, conosciuti tecnicamente col nome di *checkbox*.

Ad ogni checkbox è stata associata una tipologia di bad usability smell ed una timeline.

Abbiamo quindi inserito il form in una finestra di dialogo, realizzata con Twitter Bootstrap, visualizzabile immediatamente dopo l'utilizzo del tasto *Activate Smell Inspector*. (vedi *Figura 30*).

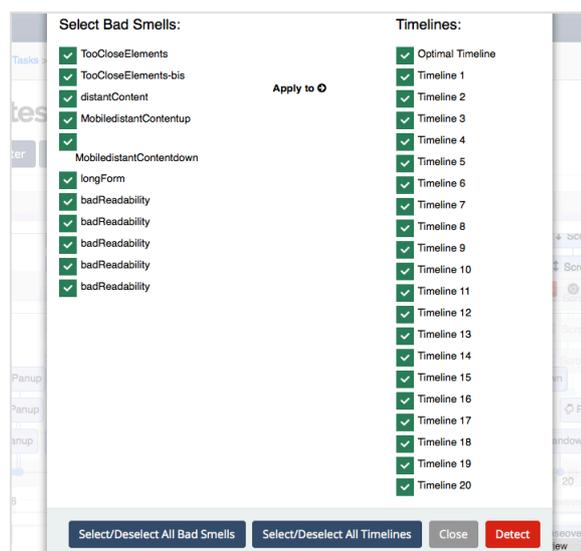


Figura 30. Finestra di dialogo per la configurazione dello smell inspector.

Come è possibile vedere dalla *Figura 30*, per velocizzare l'interazione del valutatore nella configurazione dello smell inspector sono stati inseriti, all'interno della timeline, due tasti specifici per selezionare e deselezionare tutti i bad smell e tutte le timeline per mezzo di un solo click, o di un solo tap, sul tasto dedicato.

Per concludere, il tasto *detect* e il tasto *close* sono stati implementati rispettivamente per lanciare la rilevazione dei bad usability smell e per chiudere la finestra di dialogo.

Tale finestra viene comunque chiusa automaticamente una volta individuati gli smell all'interno delle timeline.

5.4.4 La nuova modalità di confronto delle timeline

Una volta ultimata la realizzazione delle nuove timeline interattive, abbiamo deciso di implementare all'interno di MUSE una nuova funzionalità di confronto tra le varie sessioni di interazione registrate e la sequenza ottimale definita a priori dai valutatori.

Prima di descrivere nel dettaglio la nuova tipologia di confronto, è necessario fare alcune precisazioni.

Come già anticipato in precedenza, gli strumenti descritti nello stato dell'arte utilizzano un tipo di visualizzazione degli eventi basato su grafici aventi come tipologia di visualizzazione le mappe di calore.

Abbiamo deciso di non utilizzare tale rappresentazione in ragione dei limiti presentati della stessa quando applicata in un ipotetico confronto tra le sessioni.

La rappresentazione per mezzo di mappe di calore non consentiva molta interattività e, inoltre, comportava la nascita di grandi difficoltà per il valutatore nell'ambito dell'interpretazione dei dati.

Ad alimentare i nostri dubbi ha contribuito senza dubbio la funzionalità di confronto implementata precedentemente in WUP, basata sull'applicazione dell'algoritmo SAM e sull'interpretazione di dati quantitativi.

Dopo un lungo confronto, abbiamo deciso di utilizzare un approccio che portasse alla realizzazione di una metodologia di confronto tra le sessioni decisamente differente da quelle utilizzate in genere dagli strumenti di valutazione dell'usabilità.

La modalità di confronto da noi realizzata fornisce la possibilità di confrontare le sequenze di eventi con un approccio visuale. Abbiamo implementato una funzionalità in grado di sovrapporre visualmente le singole timeline alla ottimale, riuscendo a sincronizzare i comandi di navigazione su entrambe.

Per raggiungere tale obiettivo abbiamo sfruttato le potenzialità di JQUERY, CHAP LINKS LIBRARY, MOVE.JS e CSS.

Sfruttando al meglio tali librerie siamo riusciti a introdurre un buon livello di interattività anche nell'attività di confronto delle timeline, cercando favorire l'interpretazione dei dati e l'eventuale deduzione del contesto d'uso da parte del valutatore.

Intraprendere tale strada ha comportato non poche problematiche durante la fase di sviluppo di questa funzionalità.

Più precisamente, le problematiche che abbiamo incontrato nella realizzazione del confronto tra le timeline sono state le seguenti:

- Identificare le coordinate di entrambe le timeline per compiere lo spostamento della timeline oggetto del confronto sopra la timeline ottimale.
- Spostare le timeline l'una sopra l'altra.
- Allineare temporalmente le sequenze di eventi delle due timeline.
- Gestire la trasparenza delle timeline.
- Portare in primo piano, scegliendo interattivamente, l'una o l'altra timeline per avere un confronto ambidirezionale.
- Ripristinare la visualizzazione originaria a fine confronto.

La prima problematica è stata affrontata e superata attraverso l'utilizzo di JQUERY. Con la libreria di Resig abbiamo identificato all'interno del Document Object Model la timeline ottimale e quella oggetto del confronto.

Il passo successivo è stato quello di reperire, una volta individuati i due elementi, le coordinate relative alla distanza di entrambi gli elementi dal margine superiore della pagina web in cui erano situati. Ottenute le due distanze abbiamo calcolato, con una semplice sottrazione, l'ammontare dello spostamento necessario a spostare la timeline oggetto del confronto esattamente sopra la timeline ottimale. Questo tipo di approccio si è rivelato vincente, dato che JQUERY fornisce le coordinate degli elementi facendo riferimento al vertice in alto a sinistra degli stessi. Questo ha permesso, in fase di allineamento, di sovrapporre perfettamente le timeline per mezzo del metodo *translate* di MOVE.JS.

L'allineamento delle timeline è stato implementato per favorire una più semplice interazione del valutatore con le timeline sovrapposte. Volevamo evitare al valutatore di effettuare, durante un confronto, un'eventuale attività di navigazione sulle timeline disgiunte. È per questo motivo che abbiamo deciso di sincronizzarle a livello visuale e non a livello temporale.

In un primo momento avevamo valutato la potenzialità di CHAP LINKS LIBRARY di effettuare allineamenti temporali tra le varie timeline mediante il metodo *setVisibleChartRange()*. Testando tale funzionalità, ci siamo accorti che con sessioni

d'interazione troppo distanti in termini di tempi emergeva uno scenario in cui le timeline applicavano un mutamento non voluto alla propria scala temporale.

Ciò rendeva la sovrapposizione poco efficiente.

Vista tale problematica, abbiamo deciso di creare un metodo personalizzato in grado di valutare l'ammontare dello spostamento temporale della timeline oggetto dello scorrimento.

Calcolando la differenza tra il timestamp iniziale e il timestamp finale rilevato a spostamento effettuato abbiamo potuto applicare all'altra timeline uno spostamento temporale pari alla quantità di tempo definita dallo spostamento effettuato in precedenza, riuscendo a mantenere la visualizzazione più coerente e chiara.

Per quanto concerne la problematica relativa alla trasparenza delle timeline, questa è nata dopo aver intrapreso la prima visualizzazione delle timeline sovrapposte.

Ci siamo accorti che la timeline posta in primo piano copriva completamente la timeline sottostante, non permettendo la visualizzazione contemporanea di entrambe le sequenze di eventi.

Abbiamo risolto tale problematica attraverso un cambiamento dinamico del valore della proprietà CSS *opacity*. Tale proprietà assume, come insieme di valori validi, un intervallo d'interi dallo zero all'uno.

È bastato quindi impostare tale proprietà ad un valore di 0,8 sulla timeline in primo piano per riuscire a visualizzare contemporaneamente le sequenze di eventi all'interno di entrambe le timeline sovrapposte.

Tale proprietà è stata utilizzata anche nell'implementazione della funzionalità relativa allo spostamento della timeline ottimale in primo o in secondo piano.

Quando il valutatore sceglie di invertire l'ordine delle timeline, il valore di *opacity* delle due timeline viene interscambiato tra le stesse. Tale procedura è ambidirezionale.

L'ultima problematica, il ripristino della visualizzazione originaria, è stata risolta creando la procedura inversa a quelle appena descritte, visto il vantaggio di avere già disponibili i valori di spostamento e di trasparenza necessari per assolvere tale compito.

Da un punto di vista dell'interattività, la funzionalità di confronto può essere avviata per mezzo del tasto “*Overlap /detach*” di colore verde, posizionato nella testata del pannello contenente le timeline non ottimali.

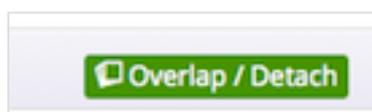


Figura 31. Pulsante per effettuare l'overlap delle timeline.

Una volta cliccato il pulsante *overlap/detach*, il medesimo cambia il proprio colore in arancio/rosso e la timeline per cui è stato attivato l'overlap effettua lo spostamento necessario a posizionarsi esattamente sopra la timeline ottimale.



Figura 32. Esempio di sovrapposizione (in corso).

Oltre al posizionamento, attivando il pulsante di overlap questo viene sostituito dal tasto *detach*, dedicato al ripristino della situazione originaria, e rende visibile il tasto *switch* attraverso cui il valutatore è in grado di portare in primo piano la timeline ottimale rispetto a quella oggetto del confronto e viceversa.

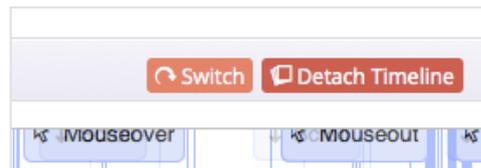


Figura 33: I tasti *switch* e *detach timeline*.

Attraverso il tasto *detach timeline* il valutatore può riportare la visualizzazione allo stadio originario. Cliccando su di esso viene reso invisibile il tasto *switch* e ripristinato il tasto di *overerlap*.

La timeline oggetto del confronto con l'ottimale torna nella propria posizione nativa, e i valori della proprietà *opacity* di entrambe le timeline vengono impostati di nuovo ad uno per eliminare la trasparenza applicata in precedenza.

5.4.5 La sezione overview

Nelle fasi di progettazione di MUSE, dopo aver realizzato le funzionalità descritte dettagliatamente nei paragrafi precedenti, una particolare attenzione è stata data all'interpretazione dei dati.

In particolare ci siamo posti l'interrogativo di come, la stessa interpretazione, potesse essere agevolata dando al valutatore una nuova funzionalità capace di fornire una visione generale dell'andamento dei singoli task della valutazione.

Questo ci ha spinti nello sviluppo della sezione *overview* dei singoli task, che abbiamo deciso di introdurre all'interno di MUSE soprattutto dopo aver riscontrato che le informazioni fornite da tale sezione erano effettivamente deducibili anche mediante un'attenta analisi delle sequenze di eventi presenti nelle timeline interattive.

Abbiamo intrapreso il primo passo verso la realizzazione della funzionalità di overview individuando quale informazione fosse davvero utile al valutatore per avere un'idea immediata circa l'andamento del task.

I dati sono stati selezionati seguendo un criterio d'importanza crescente.

Più precisamente, le informazioni scelte per implementare la sezione overview sono i seguenti:

- Riepilogo delle informazioni generali del task (nome, descrizione, URL di partenza e parametri di configurazione).
- Dati statistici sui browser utilizzati.
- Dati statistici sui sistemi operativi utilizzati
- Dati statistici su produttori e modelli di dispositivi utilizzati
- Conteggio delle occorrenze degli eventi classificati per categoria.
- Indicazioni sui custom semantic event
- Indicazioni sui Bad Usability Smell rilevati
- Indicazioni sul tempo impiegato dagli utenti per portare il task a compimento.

La pagina di overview si compone quindi di sette sezioni principali.

La sezione dedicata alle informazioni dei task è stata realizzata mediante un pannello dedicato al cui interno sono presenti le informazioni generali del singolo task, come per esempio il nome, la descrizione e l'URL di partenza (vedi *Figura 34*).



Figura 34: Pannello con le informazioni del task della sezione overview.

Abbiamo ritenuto importante fornire, con questa funzionalità, alcuni dati che potessero rappresentare per il valutatore un valido riferimento circa le dinamiche di mercato dei vari dispositivi. Per il raggiungimento di tale obiettivo abbiamo deciso di fornire informazione inerente i browser, i sistemi operativi e i modelli di device utilizzati dagli utenti durante i test utente.

La rappresentazione di tali informazioni è stata implementata attraverso un pannello contenente grafici a torta realizzati per mezzo della libreria FLOT.js (vedi *Figura 35*).

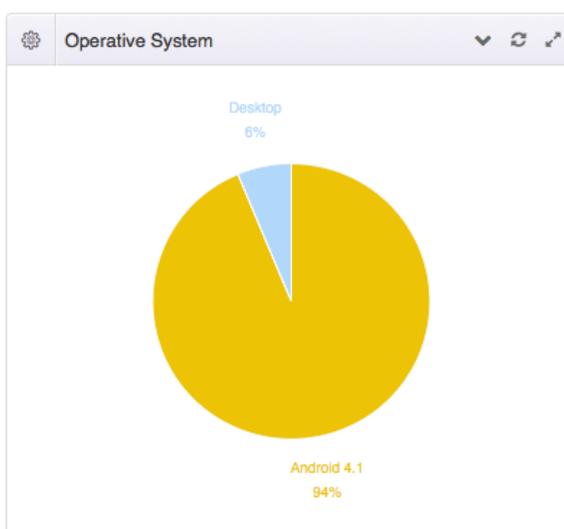


Figura 35: Esempio pannelli sezione overview- Broser, OS, Device.

La sezione relativa al sondaggio sulle occorrenze degli eventi è stata invece realizzata con l'utilizzo di un pannello contenente due box affiancati.

All'interno del primo box è visualizzabile un istogramma che fornisce al valutatore informazione visuale sul numero di eventi registrati, classificati per categoria.

Nel secondo, invece, è presente un conteggio degli eventi di ciascuna categoria.

Attraverso questo pannello informativo il valutatore può visionare quali tipologie di evento sono risultate più frequenti, avendo una prima idea circa il tipo d'interazione effettuata dagli utenti durante lo svolgimento del task.

Tale pannello è stato realizzato per mezzo della libreria FLOT.JS.

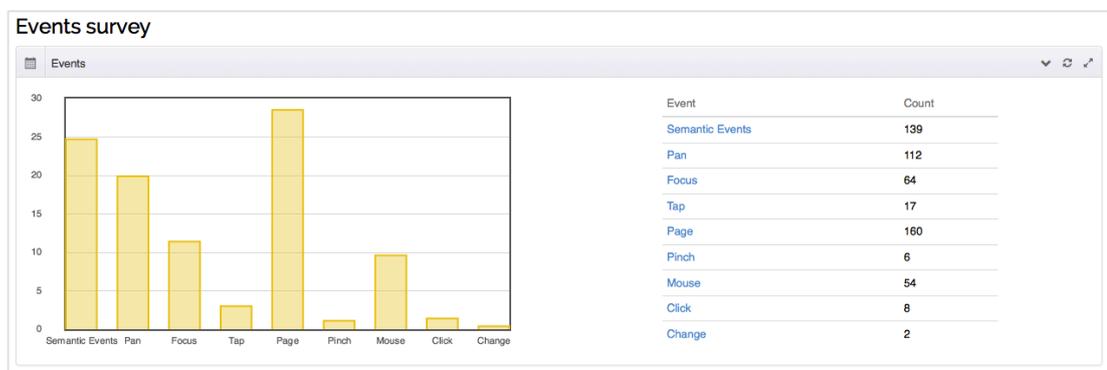


Figura 36: Pannello riassuntivo degli eventi rilevati

La sezione successiva è quella dedicata ai custom semantic event.

Ragionando sulle modalità di visualizzazione di questa categoria di eventi, abbiamo innanzitutto valutato quali fossero le informazioni prioritarie che avrebbero potuto aiutare il valutatore nell'individuazione di tali eventi nelle timeline.

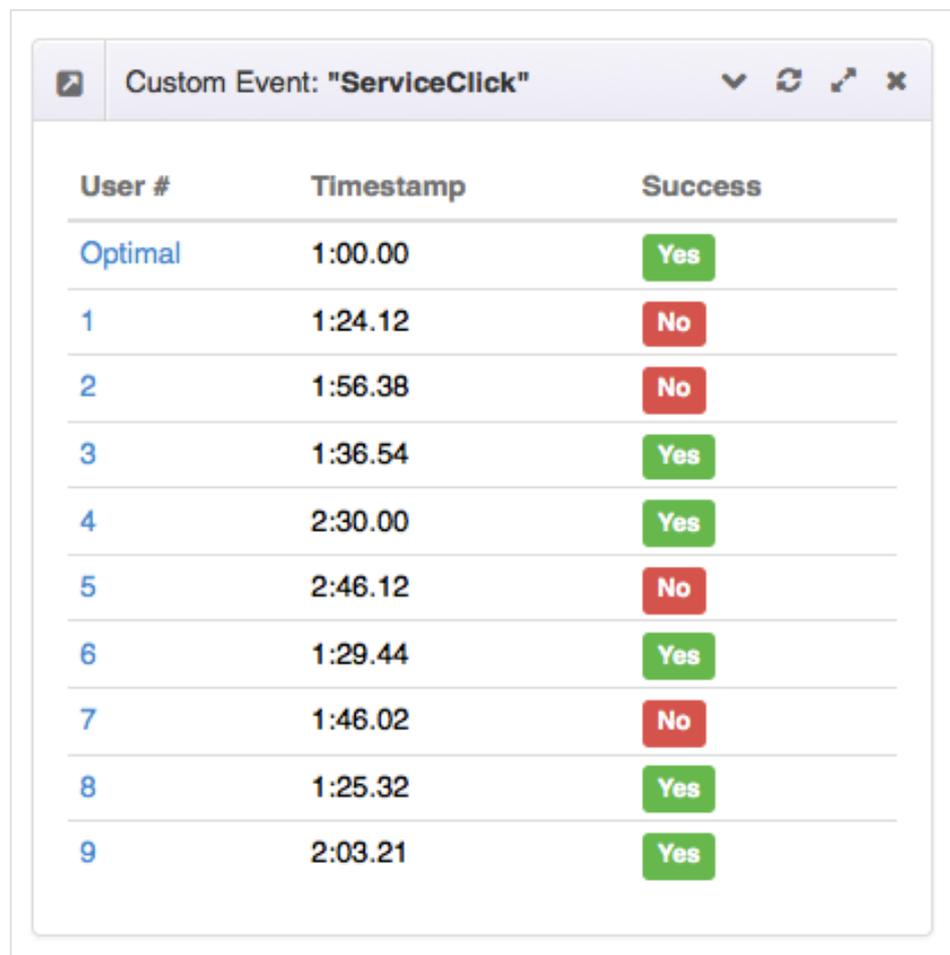
Le informazioni da noi scelte per la visualizzazione di questa sezione sono identificabili in:

- Numero identificativo della sessione.
- Tempo esatto dello scatenarsi dell'evento.
- Flag indicante se il custom event si sia verificato con successo oppure no.

Tenuto conto di questa riflessione, la visualizzazione di tali informazioni è stata realizzata mediante pannelli dedicati contenenti una tabella a tre colonne, indicanti

rispettivamente il numero identificativo della sessione (quindi della timeline), il tempo esatto dell'avvenuto evento e un'etichetta per rappresentare l'avvenuto successo o meno dell'evento semantico.

Tale etichetta, come si può notare in *Figura 37*, è stata implementata con l'accortezza di applicare un criterio cromatico. Se infatti il custom event ha avuto successo, l'etichetta viene colorata di verde e contiene la scritta "YES", mentre in caso contrario la stessa è colorata di rosso e contiene la scritta "NO".



The image shows a software interface window titled "Custom Event: 'ServiceClick'". It contains a table with three columns: "User #", "Timestamp", and "Success". The "Success" column uses green buttons for "Yes" and red buttons for "No".

User #	Timestamp	Success
Optimal	1:00.00	Yes
1	1:24.12	No
2	1:56.38	No
3	1:36.54	Yes
4	2:30.00	Yes
5	2:46.12	No
6	1:29.44	Yes
7	1:46.02	No
8	1:25.32	Yes
9	2:03.21	Yes

Figura 37: Esempio di pannello riassuntivo di un custom event.

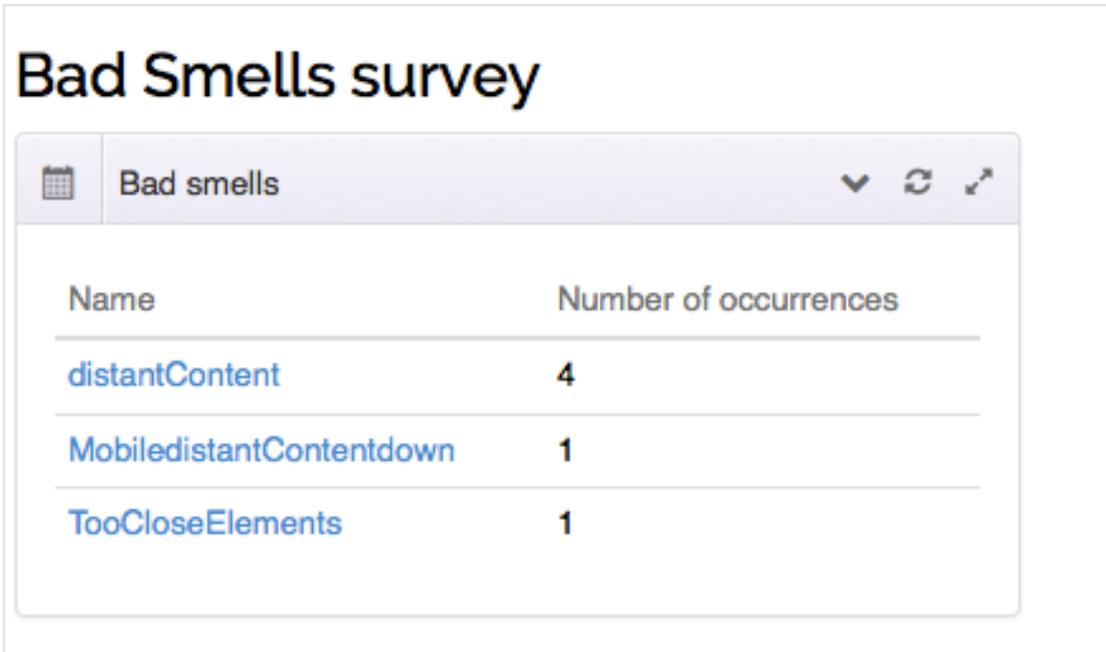
Un'altra sezione molto importante della pagina di overview è quella dedicata al riepilogo dei Bad Usability Smell rilevati all'interno del task.

Attraverso questa sezione il valutatore può ottenere a priori informazioni circa la presenza di problemi di usabilità.

I Bad Usability Smell, come già anticipato, sono infatti possibili indicatori di cattiva usabilità.

Viste le premesse, abbiamo deciso di introdurli anche nella sezione di overview per completare l'informazione relativa al task e dare una visione più ampia dell'andamento del task stesso al valutatore.

Il pannello relativo al sondaggio sui Bad Usability Smell è composto da una tabella HTML5 sviluppata su due colonne, rispettivamente indicanti il nome dello smell identificato e il suo numero di occorrenze all'interno del task.



The image shows a web interface titled "Bad Smells survey". It features a header bar with a calendar icon, the text "Bad smells", and three icons: a dropdown arrow, a refresh icon, and a share icon. Below the header is a table with two columns: "Name" and "Number of occurrences". The table contains three rows of data:

Name	Number of occurrences
distantContent	4
MobiledistantContentdown	1
TooCloseElements	1

Figura 38. Esempio di pannello raffigurante la panoramica sui Bad usability Smell della sezione Overview.

L'ultima sezione della pagina di overview è stata dedicata al tempo impiegato dai singoli utenti nel portare a compimento il task indicato. La visualizzazione dei tempi è stata effettuata per mezzo di un'istogramma della libreria FLOT.JS. La decisione di visualizzare quest'informazione attraverso questo tipo di grafico è stata presa notando che tale tipologia di rappresentazione avrebbe permesso al valutatore di farsi un'idea immediata su quale utente potesse essere più incline, esperto o meno nell'uso dell'applicazione web oggetto del test.

Tale istogramma dunque utilizza come variabili di riferimento il tempo, sull'asse delle ordinate, e le singole timeline sull'asse delle ascisse.

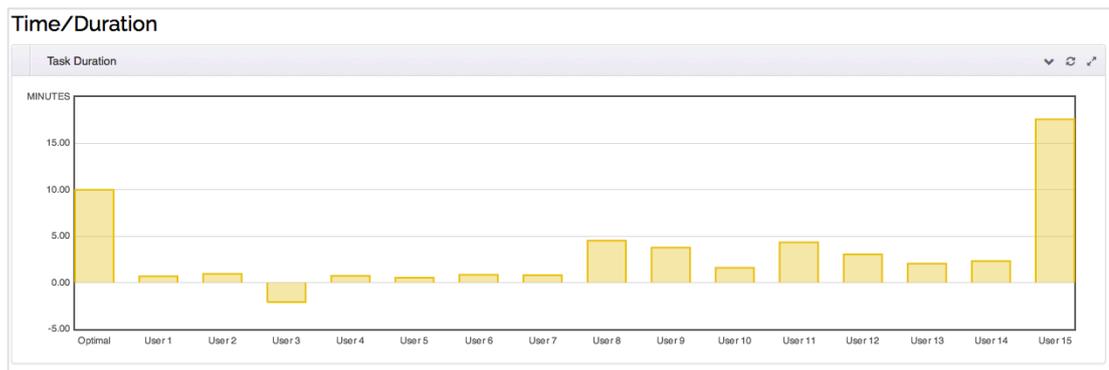


Figura 39: Esempio di un istogramma sui tempi impiegati per lo svolgimento del task.

5.4.6 La visualizzazione dell'utente in movimento

L'utilizzo delle applicazioni web in un contesto di mobilità è uno degli scenari che abbiamo deciso di valutare in fase progettuale. Analizzando tale scenario abbiamo individuato due approcci possibili per rilevare l'uso in movimento delle applicazioni web: la rilevazione del movimento mediante il sensore GPS e quella effettuata mediante l'accelerometro dei dispositivi mobili.

Il movimento, per sua natura, può essere rilevato in due contesti differenti:

il movimento *indoor* e *outdoor*. Con il termine *outdoor* si intende ogni tipo di movimento effettuato all'aria aperta, dunque all'esterno degli edifici.

Al contrario il termine *indoor* indica un tipo di movimento effettuato all'interno di edifici chiusi.

Questa distinzione è stata il punto di partenza da cui siamo partiti per il raggiungimento dell'obiettivo fissato.

Analizzando la problematica da un punto di vista delle tecnologie disponibili, il movimento *outdoor* è risultato di facile rilevazione, grazie alla tecnologia GPS presente nativamente sui dispositivi mobili di ultima generazione.

Più problematica è risultata invece la rilevazione *indoor*, dato che all'interno di edifici chiusi la tecnologia GPS ha mostrato tutti i suoi limiti, vista la mancanza di segnale riscontrata.

L'unico approccio disponibile per la rilevazione *indoor* è stato individuato nella rilevazione tramite accelerometro, anche se dai vari test effettuati in laboratorio è

emerso come l'accelerometro non possa ancora essere utilizzato in maniera efficiente.

Viste queste premesse, considerando l'approccio dell'accelerometro ancora prematuro da un punto di vista tecnologico, la nostra decisione è stata quella di effettuare, in un primo momento, la rilevazione del movimento attraverso il sensore GPS.

Il sensore GPS rileva la posizione dell'utente periodicamente, a differenza dell'accelerometro.

L'accelerometro, infatti, invia dati in maniera continua, permettendo una rilevazione continua del movimento ancora poco gestibile da un punto di vista dell'acquisizione e della memorizzazione dei dati.

In questo senso il GPS è sembrato essere la strada più semplice da percorrere per raggiungere l'obiettivo.

Allo stesso tempo la rilevazione GPS ha garantito una precisione, nella valutazione del contesto d'uso delle applicazioni web, che l'accelerometro non è stato in grado di fornire.

Mediante il sensore GPS è stato possibile ottenere coordinate precise sulla posizione dell'utente, avendo l'ulteriore possibilità di calcolare distanze percorse e velocità di movimento in maniera davvero dettagliata.

L'approccio scelto quindi si basa su una rilevazione periodica della posizione mediante la lettura di dati ricevuti dal sensore GPS. Da un punto di vista dell'implementazione, abbiamo deciso di utilizzare il metodo `getCurrentPosition()` delle HTML5 API, per la rilevazione iniziale, e il metodo `watchPosition()` per le rilevazioni periodiche successive. Lanciando questa procedura all'utente viene chiesto di attivare il rilevamento della posizione. Rilasciando il proprio consenso, l'utente dà inizio a letture periodiche dei dati forniti dal GPS, effettuate dal metodo `watchPosition()`.

Abbiamo quindi deciso di effettuare una lettura delle coordinate e di salvare queste temporaneamente fino alla lettura successiva.

Una volta ottenuto l'insieme di due rilevazioni, è stata calcolata la distanza coperta dall'utente. Nel caso in cui tale distanza abbia superato i cinquanta metri di lunghezza, l'utente è stato considerato in movimento.

Da un punto di vista della visualizzazione, è abbiamo deciso di rappresentare l'utente in movimento per mezzo di una barra verde posta nella parte bassa della timeline, in

corrispondenza dell'intervallo di tempo in cui il movimento si è verificato. Per identificare l'inizio e la fine del movimento sono state utilizzate due icone rappresentanti rispettivamente un uomo in movimento e un uomo fermo.

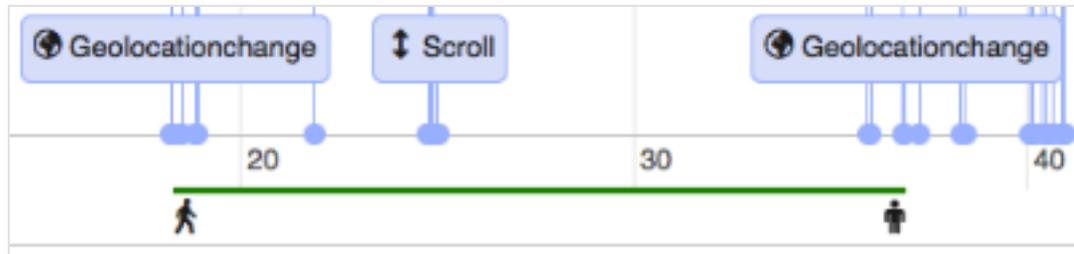


Figura 40. Esempio di visualizzazione del movimento dell'utente in MUSE.

Com'è possibile notare in *Figura 40*, all'interno della timeline sono ben riconoscibili le due rilevazioni effettuate, rappresentate da box contenenti l'evento semantico avente il nome di *Geolocationchange*. Analizzando meglio la *Figura 44* è riscontrabile che il movimento sia stato rappresentato proprio per una lunghezza pari alla distanza tra i box delle due rilevazioni e che lo stesso si sia interrotto in un momento compreso nell'intervallo definito dalle due rilevazioni.

6. Il test valutatore

6.1 Gli obiettivi del test

In seguito alla fase di sviluppo dello strumento, è stato effettuato un test utente ad un campione complessivo di quaranta persone.

L'applicazione di riferimento per il test utente è stata individuata nel sito web di *Autostrade per l'Italia S.p.a.*

Tale test è stato progettato con l'intento di far eseguire dei task mirati ai partecipanti, i quali hanno dovuto assolvere tali compiti pianificati in modo tale da avere un livello di difficoltà crescente.

Una volta ottenuti i risultati del test utente è stato deciso di effettuare un test dello strumento appositamente per i valutatori, selezionando alcuni esperti di valutazione dell'usabilità e mettendo a loro disposizione MUSE, con lo scopo di ottenere da tali soggetti un giudizio sulle potenzialità e sull'efficacia dello strumento di valutazione.

Più precisamente, dagli esperti di usabilità volevamo ottenere un feedback circa la qualità dell'informazione loro fornita dalle nuove funzionalità dello strumento, pareri ed eventuali suggerimenti per migliorare o apportare modifiche al tipo d'interazione implementato sulle timeline e infine un'opinione su come e quanto lo strumento avesse semplificato secondo gli stessi l'attività di analisi delle sequenze. Un tale obiettivo implicava il coinvolgimento dei valutatori selezionati nell'utilizzo delle funzionalità caratterizzate maggiormente da elementi interattivi, con una previa consultazione delle sezioni informative dello strumento dedicate alla presentazione globale dell'andamento dei singoli task.

I valutatori che hanno affrontato il test sono stati selezionati tenendo conto della loro competenza nella valutazione dell'usabilità. Nella selezione è stata data maggiore rilevanza ai soggetti con maggiore esperienza nell'ambito dei dispositivi mobili. Tale criterio di selezione è stato applicato in ragione delle nuove potenzialità dello strumento, dato che le stesse hanno reso possibile la rilevazione delle nuove touch gesture introdotte da tali periferiche.

6.2 La definizione del test per il valutatore

Per valutare correttamente lo scenario d'uso dello strumento, il test valutatore è stato concepito con l'obiettivo principale di mettere esperti della valutazione dell'usabilità alle prese con tutte le nuove funzionalità introdotte in MUSE.

Quello che volevamo ottenere dal test valutatore era un feedback diretto dai valutatori circa il livello di user experience che gli stessi avevano raggiunto utilizzando lo strumento. Allo stesso tempo il test valutatore ci poteva permettere di cogliere alcuni aspetti migliorabili dell'applicazione che erano passati inosservati ai nostri occhi a causa del continuo utilizzo dello strumento durante i test effettuati nella fase di sviluppo.

Dunque abbiamo deciso di redigere una breve guida introduttiva in cui tutte le funzionalità oggetto del test sono state spiegate dettagliatamente, utilizzando informazione grafica a corredo delle varie sezioni della guida stessa.

Le funzionalità oggetto del test valutatore sono state illustrate ai soggetti interessati poco prima di effettuare il test, in maniera tale da fornire loro informazioni generali sugli obiettivi prefissi dal nuovo strumento.

Più precisamente, il test è stato concepito in modo tale da far affrontare ai valutatori diversi scenari d'uso, ciascuno coinvolgente una delle funzionalità scelte.

Tra le varie funzionalità offerte da MUSE, quelle che abbiamo ritenuto più rilevanti e che abbiamo selezionato per il test valutatore sono le seguenti:

- Visualizzazione della sezione overview
- Visualizzazione delle timeline
- Filtraggio degli eventi sulle timeline
- Rilevazione dei bad smell e visualizzazione di uno screen dump
- Confronto tra timeline ottimale e timeline generica

Analizzando nel dettaglio la struttura del test è possibile descrivere in maniera più dettagliata ciò che volevamo ottenere dai valutatori.

La visualizzazione della pagina di overview è stata scelta, tra le varie funzionalità, per ottenere un'informazione ben specifica. Volevamo conoscere la loro impressione sulla chiarezza e sulla rilevanza dei dati forniti, cercando di sapere da loro se tale funzionalità poteva essere di supporto nella comprensione dell'andamento del task.

La visualizzazione delle timeline è stata selezionata per conoscere dai valutatori le loro impressioni, scaturite da un primo impatto con l'interfaccia, circa la nuova modalità di visualizzazione delle timeline. Di questa funzionalità volevamo conoscere quanto avessero ritenuto usabili ed utili le nuove timeline interattive, ai fini di una corretta valutazione dell'usabilità

La successiva funzionalità che abbiamo voluto far testare ai valutatori è stata il filtraggio degli eventi. Tale funzionalità è stata scelta per ottenere un feedback sulla possibile semplificazione della visualizzazione apportata da questo strumento. Volevamo avere un giudizio dai valutatori su come e quanto il filtraggio degli eventi potesse contribuire ad una più chiara e precisa lettura dei dati e ad una migliore interpretazione degli stessi.

L'identificazione e la visualizzazione dei bad usability smell è stata forse la funzionalità più rilevante che volevamo i valutatori testassero. Su questa feature volevamo conoscere quanto la stessa potesse essere di loro aiuto nell'individuazione di problemi di usabilità e quanto, secondo il loro giudizio, potesse essere considerata utile per garantire un'eventuale minor perdita di tempo nell'attività di valutazione. Oltre a questo, dal test volevamo ottenere informazioni utili sulla visualizzazione adottata per la rappresentazione di tali indicatori di cattiva usabilità, cercando di cogliere suggerimenti utili per migliorare ulteriormente tale rappresentazione.

L'ultima funzionalità che abbiamo fatto provare ai valutatori è il confronto visuale delle timeline. Abbiamo spinto i valutatori ad effettuare la sovrapposizione di una timeline generica sopra la timeline ottimale. Una volta effettuata la sovrapposizione, abbiamo cercato di ottenere da loro un feedback relativo al nuovo approccio di confronto implementato all'interno di MUSE. Più precisamente, volevamo sapere se tale modalità di confronto fosse da loro ritenuta più valida e vantaggiosa rispetto ad una modalità di confronto basata su dati quantitativi.

6.3 *La stesura del questionario*

Dopo aver effettuato il test è stato chiesto ai valutatori di compilare un questionario elettronico, in maniera tale da avere dati conservabili a lungo termine.

Abbiamo deciso di comporre il questionario con trentotto domande mirate ai punti cruciali del test che i valutatori hanno effettuato.

Il questionario è stato redatto per mezzo della suite messa a disposizione dal servizio Google Drive di *Google Inc*, inserendo tipologie di risposta differenti a seconda del tipo di domanda effettuata.

Per domande da cui necessitavamo un giudizio in scala, sono state utilizzate le domande relative dei moduli di Google, utilizzando per la definizione dei giudizi un intervallo di valori dall'uno al sette.

Per le risposte aperte è stata invece scelta la modalità di risposta per mezzo di aree di testo in grado di permettere all'utente di generare risposte più lunghe.

Proprio come la realizzazione del test, il questionario elettronico è stato realizzato prendendo come punti di focus le funzionalità di MUSE che dovevano essere testate.

Il questionario presenta nella sua prima parte alcune domande personali per il valutatore. Abbiamo voluto conoscere il sesso del valutatore, la sua età e quali competenze avesse circa l'utilizzo di linguaggi di programmazione e di markup. Abbiamo deciso di porre quest'ultima domanda per poter capire quale competenza avessero i valutatori nell'uso delle tecnologie che erano alla base del funzionamento del programma. Inoltre, con questo quesito, valutatori competenti avrebbero potuto darci anche suggerimenti tecnici per migliorare alcune funzionalità presenti nello strumento.

Ai valutatori è stato chiesto, tra le varie domande personali, di indicare se avessero mai utilizzato strumenti automatici per la valutazione dell'usabilità e, in caso affermativo, se ricordassero quale tipologia di acquisizione dati utilizzassero tali strumenti per immagazzinare informazioni sul comportamento dell'utenza. Questa informazione era per noi molto rilevante, dato che forniva un dato importante sugli strumenti utilizzati in precedenza dai valutatori.

Quanto alle domande inerenti le funzionalità di MUSE, queste sono state divise all'interno del questionario in sezioni differenti:

- Giudizio generale sull'interfaccia amministratore
- Valutazione della sezione overview
- Valutazione delle timeline interattive
- Valutazione della funzionalità di confronto delle timeline
- Valutazione della funzionalità di filtraggio degli eventi
- Valutazione delle funzionalità di identificazione e rappresentazione dei Bad Usability Smell

Per il primo punto è stato chiesto ai valutatori come fosse loro sembrata la nuova interfaccia dello strumento ad un primo impatto. In particolare è stato loro chiesto se, a loro giudizio, gli elementi all'interno dell'interfaccia fossero ben presentati.

Circa la sezione overview ai valutatori è stato chiesto in primis se ritenessero sufficiente l'informazione fornita da tale funzionalità. Continuando abbiamo cercato di ottenere un giudizio sull'utilità di questa feature per l'eventuale comprensione dell'andamento generale del task, chiedendo inoltre quanto ritenessero rilevante e attinente a tale compito l'informazione rappresentata.

Sono state poste domande anche sul livello d'interattività delle timeline. Ai valutatori abbiamo chiesto di valutare se ritenessero idoneo il livello di interazione fornito dal nuovo strumento. In particolare, per questa sezione ci siamo soffermati sulle funzionalità relative alla manipolazione della visualizzazione delle timeline (zoom in e zoom out) e su quelle relative alla navigazione delle sequenze di eventi rappresentate.

La sezione relativa alla nuova funzionalità di confronto è stata senza ombra di dubbio quella con il maggior numero di quesiti. Ai soggetti abbiamo posto domande cercando risposte inerenti il livello di usabilità e di utilità della nuova procedura. Nel caso in cui un valutatore ritenesse più vantaggioso il nuovo metodo di confronto delle timeline rispetto ad un approccio basato su dati quantitativi, abbiamo deciso di fare motivare le risposte. Le risposte aperte sono state di grande aiuto, in casi come questo, a noi sviluppatori. Da tali risposte siamo riusciti a comprendere alcuni aspetti migliorabili della funzionalità.

Allo stesso modo è stata gestita la sezione del questionario relativa alla valutazione dello

strumento dedicato al filtraggio degli eventi. Abbiamo chiesto di nuovo quanto ritenessero usabile tale strumento, quali fossero secondo loro le criticità riscontrate nell'utilizzo dello strumento e infine quanto ritenessero lo strumento utile ai fini di una più chiara interpretazione del comportamento degli utenti.

L'ultima sezione del questionario è stata incentrata sull'identificazione e sulla rappresentazione dei bad usability smell. Ai valutatori abbiamo chiesto una valutazione sull'utilità e sull'usabilità del nuovo strumento, in particolare per ottenere un feedback sul come e quanto tale funzionalità possa essere considerata efficiente per individuare problemi di usabilità spendendo un minor quantitativo di tempo utile. Su questa sezione sono state poste, in maniera più dettagliata, anche domande sulle fasi d'interazione precedenti e posteriori l'attivazione della procedura di rilevazione.

6.4 I risultati del test valutatore

Abbiamo sottoposto al test sette valutatori esperti di usabilità. Più precisamente, il test è stato effettuato da sei uomini e una donna, con un'età media rilevata uguale a 35,7 anni.

Tutti hanno dichiarato di avere esperienza nell'ambito della valutazione dell'usabilità e competenze su alcuni dei linguaggi di programmazione e di markup più utilizzati (Java, XML, HTML5). Alcuni di questi conoscevano WUP, lo strumento di valutazione dell'usabilità da cui MUSE ha preso forma.

Dalla lettura dei questionari compilati dopo l'esecuzione del test sono emersi molti dati utili e alcune criticità dell'applicazione.

Analizzando le risposte fornite dai valutatori, abbiamo raccolto un giudizio complessivamente positivo circa le funzionalità dello strumento che abbiamo sviluppato.

In particolare, un buon giudizio è stato attribuito alla nuova interfaccia lato amministratore del programma. I valutatori hanno ritenuto la presentazione dei contenuti sufficientemente chiara, attestando un valore medio, in una scala da uno a sette, pari a 5,2.

Come ti è sembrata, al primo impatto, la nuova interfaccia lato amministratore di Muse ?

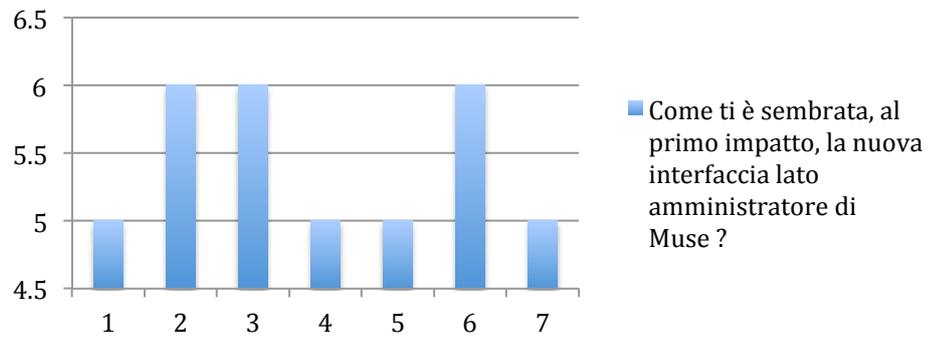


Figura 41.

Ritieni che gli elementi siano ben presentati all'interno dell'interfaccia lato amministratore di Muse ?

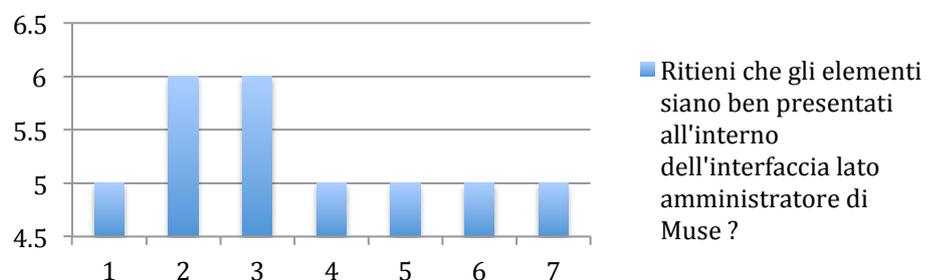


Figura 42.

La sezione overview è stata giudicata molto positivamente dai valutatori. Com'è possibile notare dalle *Figure 43 e 44* i valutatori hanno ritenuto molto utile la sezione *overview* per avere un primo panorama sull'andamento generale dei task. Dalla *Figura 44* emerge anche il fatto che i valutatori abbiano ritenuto molto indicative le informazioni inserite all'interno di *overview*.

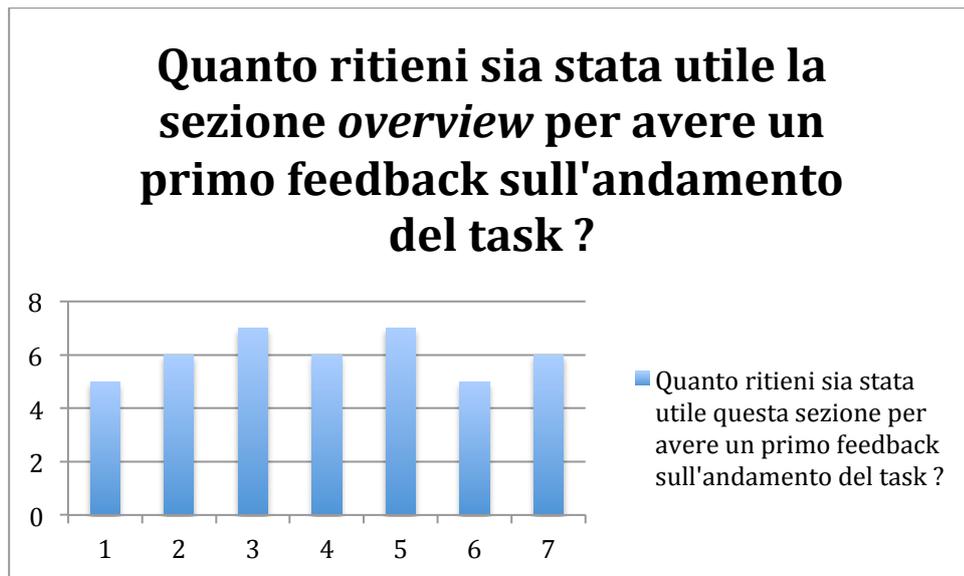


Figura 43.



Figura 44.

Alla sezione *overview* non sono tuttavia mancate alcune critiche.

I valutatori, hanno criticato il fatto che nella sezione inerente le informazioni generali del task non fosse presente il numero totale di sessioni d'interazione registrate. In effetti, dopo aver ricevuto tale critica, abbiamo constatato che per sapere il numero totale delle sessioni fosse necessario scorrere fino alla sezione *time/duration*.

Alcuni valutatori hanno segnalato poi una mancata dichiarazione degli obiettivi di ogni singola sotto-sezione di overview, ed hanno suggerito di inserire nella sezione relativa alle informazioni generali del task uno screenshot della pagina web oggetto del task.

Inoltre, tra i suggerimenti forniti per questa sezione, uno dei valutatori ha consigliato, nel caso di task caratterizzati da molte sessioni registrate, di raggruppare le stesse nella sezione *time/duration* prendendo come discriminante una serie di intervalli di tempo stabiliti.

La visualizzazione degli eventi delle timeline è risultata ai valutatori tutto sommato sufficientemente chiara (vedi *Figura 45*). Solo uno dei valutatori ha rivolto una critica alle timeline, critica che può però considerarsi concettuale. Tale valutatore riteneva non fosse corretta la presenza di una singola timeline ottimale. Secondo lo stesso valutatore il modello di task può infatti prevedere più cammini ideali per il raggiungimento dello stesso obiettivo. Avrebbe quindi preferito fossero presenti anche altre timeline ottimali indicanti sequenze d'interazione ideali in aggiunta a quella fornita a priori da noi sviluppatori.

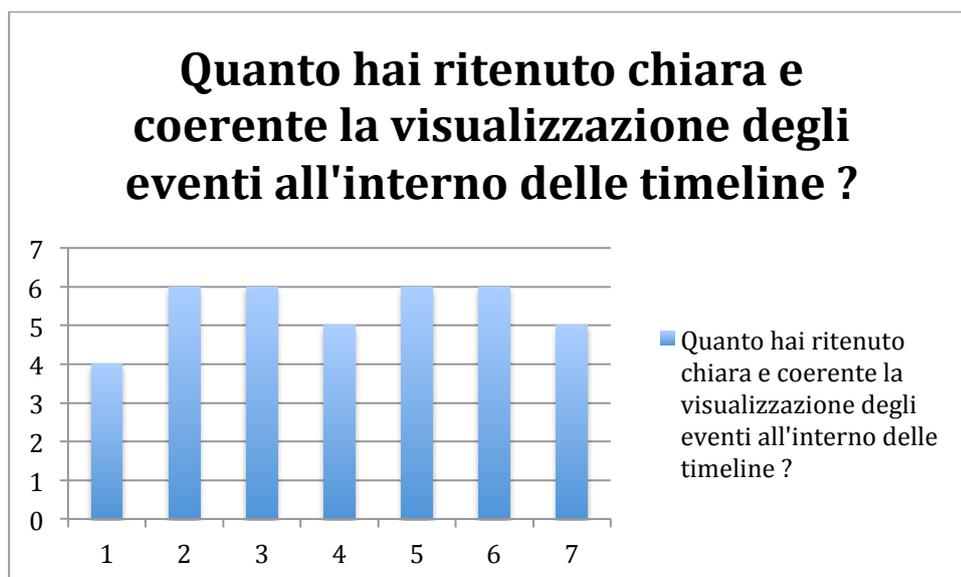


Figura 45.

Tutti i valutatori, ad eccetto di uno, hanno ritenuto il livello di interattività delle timeline davvero buono, anche se migliorabile in alcune funzionalità.

Sono stati ritenuti davvero utili ed usabili gli screenshot relativi alle pagine web navigate dagli utenti. I valutatori, su questa funzionalità, hanno riferito nel questionario che tale funzionalità è in grado di chiarire molto la posizione in cui si trovava l'utente durante il test. Inoltre hanno sostenuto che gli screenshot possono essere utili al valutatore stesso per capire quale tipo d'interazione potrà aspettarsi all'interno delle sequenze d'interazione rilevate.

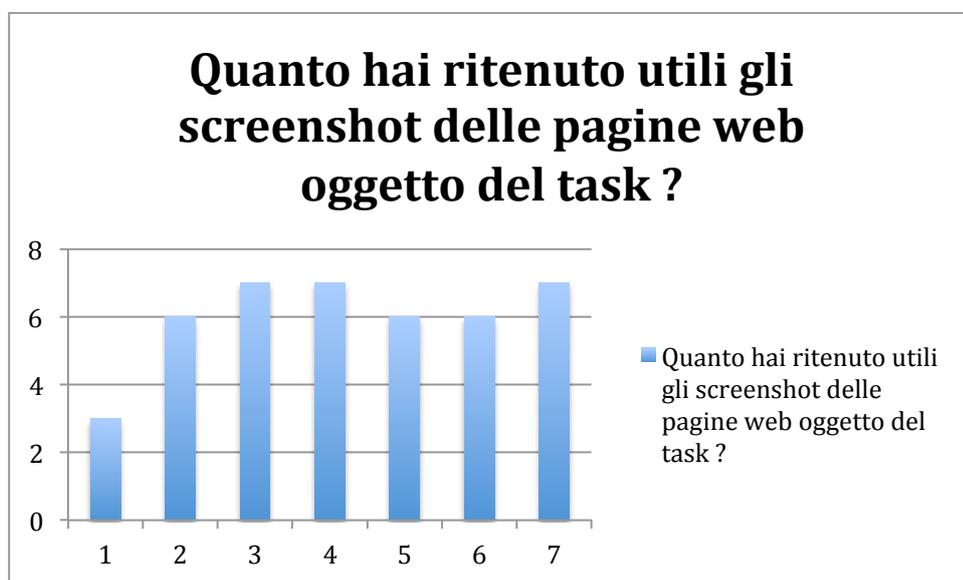


Figura 46.

Altre funzionalità su cui i valutatori hanno espresso un loro parere sono le funzioni dedicate alla navigazione e alla manipolazione delle timeline. Com'è possibile rilevare in *Figura 47* e *Figura 48*, alcuni valutatori hanno dato un giudizio appena sufficiente ai controlli per la navigazione delle timeline. Dalle risposte fornite, è emerso che nei casi con timeline popolate da molti eventi lo spostamento fosse leggermente lento e poco performante.

Quanto alle funzionalità di manipolazione della timeline, l'attività di zoom è stata ritenuta utile ed usabile. Anche questa funzionalità non è stata esente da alcune osservazioni. Alcuni valutatori si sono sentiti un po' spiazzati a causa del cambiamento automatico della scala temporale, nonostante il miglioramento notato circa la leggibilità delle sequenze di interazione. Uno dei valutatori, circa questa

funzionalità, ha osservato che i tasti + e - potrebbero trarre in inganno. Dunque avrebbe preferito avere i classici tasti + e - inseriti all'interno di un'icona raffigurante una lente d'ingrandimento.

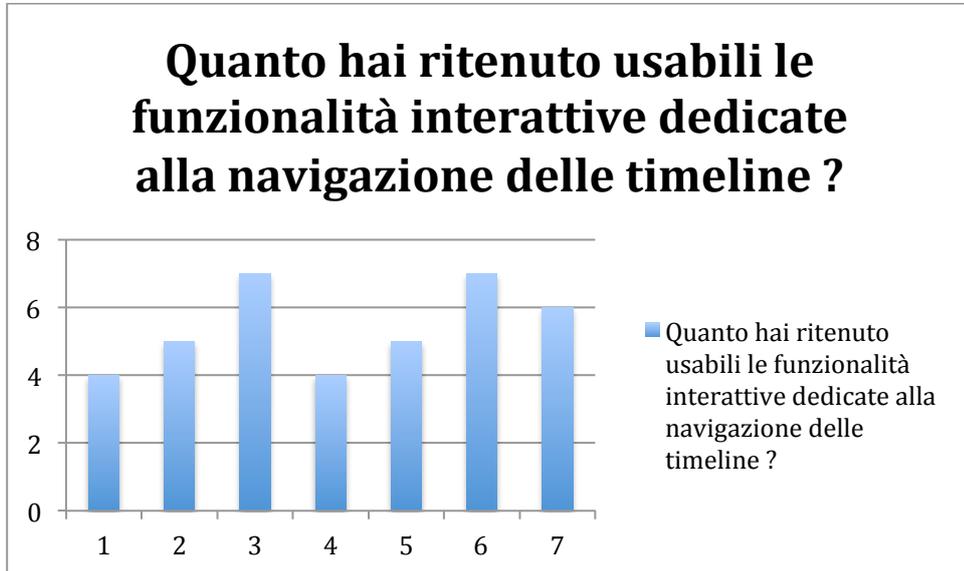


Figura 47.



Figura 48.

Il confronto visuale per mezzo della sovrapposizione delle timeline è la funzionalità che è stata più discussa dai valutatori. Dai giudizi rilevati dal questionario, nel gruppo dei valutatori si sono create due correnti di pensiero che rispettivamente ritenevano tale funzionalità molto innovativa ed utile e poco usabile.

Tra le critiche proposte, uno dei valutatori ha ritenuto che un confronto di questo tipo potrebbe portare a valutazioni poco corrette, per esempio nel caso in cui si verificano sequenze corrette da un punto di vista del modello del task ma non corrette da un punto di vista del confronto visuale. Secondo il valutatore infatti non è detto che il tempo possa essere in ogni caso il discriminante corretto per effettuare un confronto. La tipologia dell'evento stesso, secondo lo stesso valutatore, sembrava essere un riferimento più valido.

Quanto ai giudizi positivi altri valutatori hanno ritenuto che la sovrapposizione di due timeline sincronizzate fosse certamente uno strumento per effettuare un confronto più immediato e vantaggioso, specialmente rendendo lo strumento, magari in futuro, in grado di effettuare allineamenti delle sequenze non in base al tempo ma anzi in base ad un determinato tipo di evento o addirittura stabilendo relazioni semantiche tra le azioni.

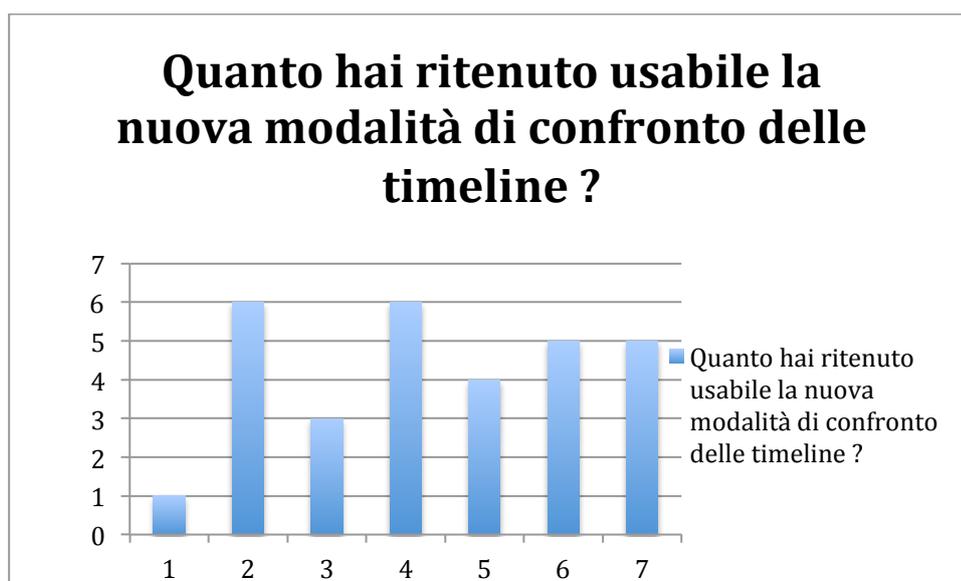


Figura 49.



Figura 50.

La funzione di filtraggio degli eventi ha riscosso moltissimi giudizi positivi da un punto di vista dell'utilità. Una critica per questa funzionalità è stata fatta da più valutatori. Il filtro, così com'è stato progettato, nasconde gli eventi che il valutatore seleziona per mezzo del menu a tendina dedicato. I valutatori avrebbero voluto che il filtro funzionasse inversamente. Avrebbero preferito che gli eventi scelti per mezzo del menu a tendina fossero quelli che il valutatore intendeva far rimanere visibili all'interno delle timeline. Tale richiesta è alla base dei voti poco alti all'usabilità della funzionalità di filtraggio.

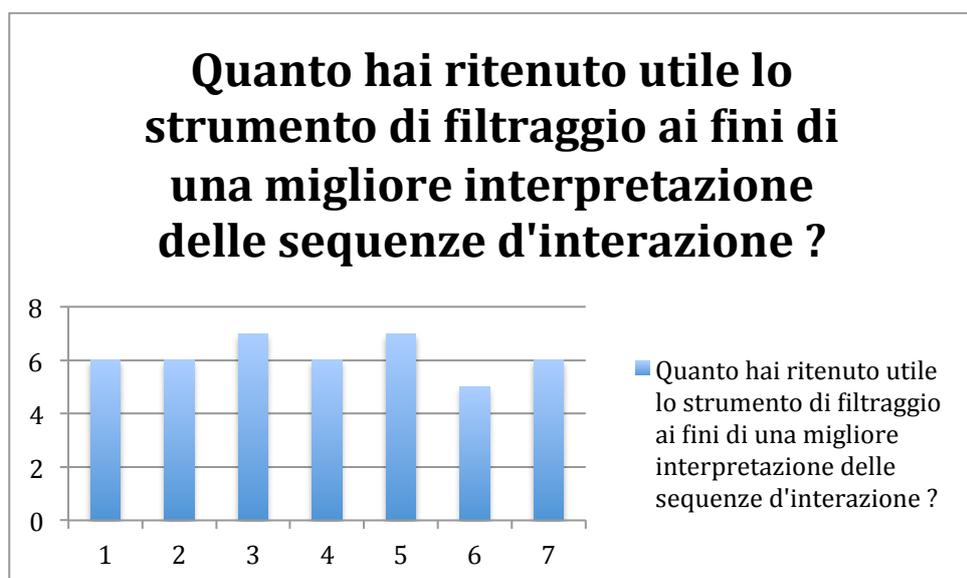


Figura 51.

Quanto hai ritenuto usabile lo strumento di filtraggio ?

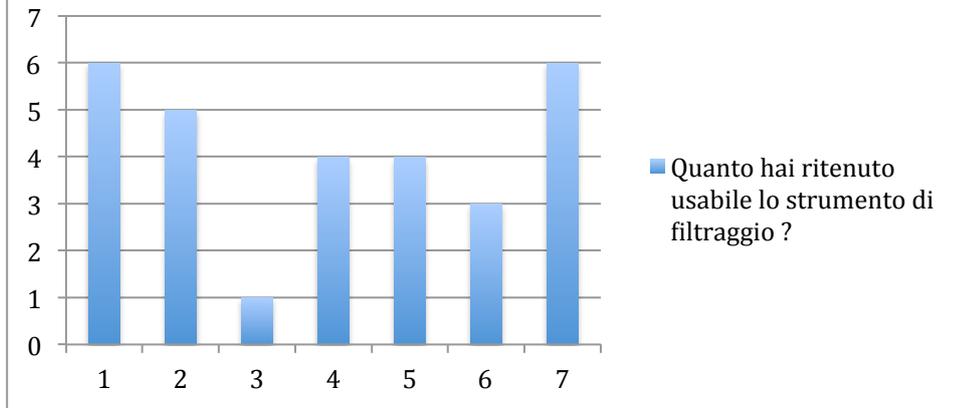


Figura 52.

7. Conclusioni

Questa tesi aveva come obiettivo principale la realizzazione di uno strumento di valutazione dell'usabilità in grado di facilitare al valutatore il compito dell'identificazione dei problemi d'usabilità delle applicazioni web.

La realizzazione di questo strumento è stata portata avanti in primis sviluppando le funzionalità dell'applicazione dedicate all'acquisizione dati. Successivamente è stata realizzata la parte dello strumento comprendente la presentazione dei dati e gli strumenti di analisi avanzata a disposizione del valutatore.

L'esigenza alla base del lavoro svolto era quella di fornire uno strumento di supporto ai valutatori di usabilità in grado di aiutare gli stessi nelle loro più frequenti attività, introducendo nuove funzionalità caratterizzate da un livello d'interattività maggiore rispetto a quella fornita dagli altri applicativi. Dunque volevamo che il valutatore, per mezzo delle funzionalità di MUSE, riuscisse ad effettuare le valutazioni di usabilità riducendo i tempi di lavoro grazie ad un'informazione presentata in maniera logica, chiara e coerente con lo scenario d'uso delle applicazioni rappresentato dalle sequenze di dati.

In più, grazie all'interattività delle nuove funzionalità, volevamo che il valutatore disponesse, in breve tempo, di una buona base di dati raffinati su cui poter condurre uno studio.

Nello sviluppo dello strumento abbiamo incontrato non poche problematiche, che fortunatamente grazie all'evoluzione delle risorse tecnologiche siamo riusciti a risolvere nella maggior parte dei casi. Tuttavia, alcune limitazioni non ci hanno permesso di acquisire i dati di elementi basati su particolari tecnologie, come per esempio alcuni plugin utilizzati per la visione di video su pagine web (Flash, Silverlight).

Le nuove funzionalità introdotte da MUSE, come si può notare dai risultati del test valutatore, sono state giudicate dalla maggior parte dei valutatori come utili ed usabili, anche se in alcuni casi gli stessi hanno individuato alcune mancanze e limiti che, se colmati, potranno migliorare ulteriormente lo strumento.

In particolare, dalla sezione di overview, non è risultato immediato per i valutatori capire quanti utenti avessero effettuato il task. Uno degli sviluppi futuri potrebbe

essere proprio l'introduzione di questo dato nel pannello della sezione overview dedicato alla visualizzazione delle informazioni generali dei singoli task.

Un altro aspetto migliorabile di questa sezione, a parere di uno dei valutatori, potrebbe essere la parte dedicata alla rappresentazione del tempo impiegato dagli utenti nell'effettuare il task. Alcuni valutatori hanno sottolineato che potrebbe essere introdotto un dato relativo al tempo minimo e massimo impiegato, prendendo questi due valori dall'insieme delle sessioni registrate. In futuro, tale informazione, potrebbe essere inserita a corredo del grafico rappresentante le tempistiche impiegate dagli utenti nello svolgimento del task e potrebbe essere effettuata una aggregazione dei dati prendendo come valori discriminanti degli intervalli di tempo precisi.

Per ciò che concerne le timeline interattive, alcuni valutatori hanno giudicato i tempi di caricamento delle stesse lenti nei task caratterizzati da molte sessioni di interazione registrate. In questo senso, uno degli sviluppi futuri potrebbe essere individuato in un'ottimizzazione degli script alla base di questa funzionalità. Va precisato però che la lentezza nel caricamento delle timeline non può essere associata solo ed esclusivamente allo strumento di valutazione. Variabile di grande rilevanza, in questo caso, è certamente la qualità della connessione con cui lo strumento è stato utilizzato.

La nuova modalità di confronto è invece una delle nuove funzionalità che abbiamo introdotto sapendo che un approccio puramente visuale avrebbe potuto essere rischioso da un punto di vista del giudizio dei valutatori. Alcuni valutatori hanno indicato alcune difficoltà nell'interpretazione dei dati nei casi caratterizzati da timeline troppo popolate. Tale funzionalità potrebbe essere migliorata evitando la sovrapposizione della timeline ottimale con una delle timeline generiche e posizionando la timeline oggetto del confronto subito sotto la timeline ottimale.

Considerando però che da alcuni valutatori l'idea della sovrapposizione è stata giudicata positivamente, una soluzione potrebbe essere l'introduzione di una finestra di dialogo che permettesse loro di scegliere quale tipo di confronto attivare, dando come possibilità di scelta sia la sovrapposizione che l'affiancamento delle timeline.

Alcuni suggerimenti dei valutatori hanno coinvolto anche la funzionalità di filtraggio degli eventi. Alcuni valutatori hanno richiesto espressamente una procedura di selezione degli eventi inversa a quella proposta dalla funzionalità. Tale miglioria, potrebbe essere introdotta in breve tempo modificando il codice dedicato alla gestione del menu a tendina presente proprio per assolvere questo compito.

MUSE, dunque, è un'applicazione dalle grandi potenzialità. Viste le difficoltà incontrate, possiamo sostenere che realizzare tale strumento sia stata una bella sfida e una grande occasione per aumentare le nostre competenze professionali.

MUSE è stato realizzato in modo tale da renderlo flessibile, per i possibili sviluppi ed evoluzioni future che abbiamo appena citato.

Tra questi, abbiamo valutato la possibilità di utilizzare, per sviluppi futuri, algoritmi di Data Mining per attuare a priori un ulteriore affinamento delle sequenze, garantendo al valutatore una visualizzazione ancora più chiara e coerente.

8. Appendice A: le domande del questionario

A seguire sono presentate le domande del questionario proposto, dopo aver effettuato il test, ai valutatori selezionati:

1. Sesso
2. Età
3. Hai esperienza nella programmazione web ?
4. Se sì, quali di questi linguaggi di programmazione e/o markup conosci meglio ?
5. Prima di questo test, hai mai utilizzato strumenti automatici per la valutazione dell'usabilità ?
6. In caso affermativo, puoi indicarci precisamente quali strumenti hai usato ?
7. Nel caso tu li abbia indicati, ricordi per caso quale approccio utilizzavano tali strumenti per effettuare la raccolta dei dati del comportamento dell'utente ?
8. Come ti è sembrata, al primo impatto, la nuova interfaccia dello strumento ?
9. Ritieni che gli elementi siano ben presentati all'interno dell'interfaccia lato amministrazione ?
10. Ritieni sufficienti le informazioni fornite nella sezione *overview* ?
11. Quanto ritieni sia stata utile questa sezione per avere un primo feedback sull'andamento di un singolo task ?
12. Quanto ritieni siano indicativi e rilevanti i dati scelti per questa sezione ?

13. Ritieni che in *overview* la presentazione dei dati sia stata sufficientemente chiara ?
14. Hai eventuali indicazioni da darci per migliorare la sezione di overview dei task ?
15. Se sì, quali sarebbero tali indicazioni ?
16. Quanto hai ritenuto chiare e coerente la visualizzazione degli eventi all'interno delle timeline ?
17. Quanto hai ritenuto utili gli screenshot delle pagine web oggetto del task ?
18. Quanto hai ritenuto usabili le funzionalità interattive dedicate alla navigazione delle timeline ?
19. Quanto hai ritenuto usabili le funzionalità di zoom dedicate alla manipolazione della visualizzazione delle timeline ?
20. Hai ritenuto sufficiente il livello di interattività fornito dalle timeline ?
21. In caso negativo, hai suggerimenti da dare per il miglioramento delle timeline ?
22. Quanto hai ritenuto usabile la nuova modalità di confronto delle timeline ?
23. Pensi che il confronto visuale sia stato più vantaggioso rispetto ad uno eseguito con dati quantitativi ?
24. In caso affermativo, potresti motivare la tua risposta ?
25. Quanto hai ritenuto usabile lo spostamento all'unisono delle due timeline dopo l'avvenuta sovrapposizione ?

26. Hai eventuali suggerimenti per il miglioramento di questa funzionalità ?
27. Quanto hai ritenuto utile lo strumento di filtraggio ai fini di una migliore interpretazione delle sequenze d'interazione ?
28. Quanto hai ritenuto usabile lo strumento di filtraggio ?
29. Ritieni che la funzione di filtraggio possa essere migliorata ?
30. In caso affermativo, potresti motivare la tua risposta ?
31. Sei riuscito ad identificare dei bad usability smell durante il test ?
32. In caso affermativo, ritieni l'evidenziazione dei bad smell sulle timeline ben distinta rispetto agli eventi generici ?
33. Quanto hai ritenuto usabile la funzionalità dedicata all'identificazione dei bad usability smell ?
34. Quanto ritieni possa essere utile tale strumento per l'identificazione di problemi di usabilità ?
35. Quanto ritieni utile la rappresentazione dei bad usability smell all'interno degli screenshot delle pagine web oggetto del task ?
36. Hai trovato difficoltà nell'attivare la finestra di dialogo per la visualizzazione dei bad usability smell all'interno degli screenshot ?
37. Ritieni che lo strumento per la visualizzazione dei bad usability smell possa essere migliorabile ?
38. In caso affermativo, potresti motivare la tua risposta ?

Bibliografia

- [1] Santana, Baranauskas, *WELFIT: A remote evaluation tool for identifying Web usage patterns through client-side logging*, 2014.
- [2] Grigera J., Garrido A., Rivero J.M., *A Tool for Detecting Bad Usability Smells in an Automatic Way*, Springer International Publishing, 2014
- [3] Grigera, Garrido, Rivero, *Refactoring for usability in web applications*, IEEE, 2011
- [4] Bateman S., Gutwin C., Osgood N., McCalla G., *Interactive Usability Instrumentation*. ACM New York, NY, USA 2009
- [5] Paganelli, Paternò, *Tools for remote usability evaluation of web applications through browser logs and task models*, Physionomics Society, 2003
- [6] Rauch M., *Mobile Documentation: Usability Guidelines, and Considerations for Providing Documentation on Kindle, Tablets, and Smartphones*, Professional Communication Conference (IPCC), 2011
- [7] Ling Bao, Intille S., *Activity Recognition from User-Annotated Acceleration Data*. Springer Berlin Heidelberg, Germany, 2004
- [8] Das S., Green L., Perez B., Murphy M., *Detecting User Activities using the Accelerometer on Android Smartphones*, The Team for Research in Ubiquitous Secure, 2010
- [9] Kwapisz J., Weiss G.M., Moore S.A., *Activity Recognition using Cell Phone Accelerometers*, ACM New York, NY, USA, 2011
- [10] Carta T., Paternò F., *A Tool for Supporting Remote Usability Evaluation of Web Sites*, Springerlink, 2011
- [11] Ivory M.Y., Hearst M.A., *The state of the art in automatic usability evaluation of user interfaces*, ACM Computing Surveys, 2001
- [12] Nielsen J., Budiu R., *Mobile usability*, New Riders Press, CA, USA, 2012
- [13] Marcotte E., *Responsive Web Design. A book apart*, New York, NY, 2010
- [14] Harms P, Grabowski J., *Usage-Based Automatic Detection of Usability Smells*, Springer Berlin Heidelberg, Germany, 2014
- [15] Nebeling M., Speicher M., Norrie M.C., *W3Touch: Metrics-based Web Page Adaptation for Touch*, ACM New York, NY, USA 2013

- [16] Buschek D., Auch A., Alt F., *A Toolkit for Analysis and Prediction of Touch Targeting Behaviour on Mobile Websites*, ACM New York, NY, USA 2015
- [17] Dündar B., Arsoy S., Yumusak N., *Guided-Based Usability Evaluation On Mobile Websites*, ICWE conference, Rome, Italy, 2013

Siti web consultati

- [1W] NNGROUP
Nielsen, Jacob, *Usability 101: Introduction to Usability*, 4 Gennaio 2012
<http://www.nngroup.com/articles/usability-101-introduction-to-usability/>
(visitato il 17 ottobre 2015)
- [2W] NNGROUP
Budiu, Raluca, *The state of Mobile User Experience*, 22 Marzo 2015
<http://www.nngroup.com/articles/mobile-usability-update/>
(visitato il 17 ottobre 2015)
- [3W] NNGROUP
Nielsen, Jacob, *Mobile Usability First findings*, 22 Luglio 2009
<http://www.nngroup.com/articles/mobile-usability-first-findings/>
(visitato il 17 ottobre 2015)
- [4W] NNGROUP
Schade, Amy, *Responsive Web Design (RWD) and User Experience*,
22 Luglio 2009
<http://www.nngroup.com/articles/mobile-usability-first-findings/>
(visitato il 17 ottobre 2015)

- [5W] JQUERY
<http://www.jquery.com>
(visitato il 17 ottobre 2015)
- [6W] CRAZYEGG
<http://www.crazyegg.com>
(visitato il 17 ottobre 2015)
- [7W] CLICKTALE
<https://www.clicktale.com>
(visitato il 17 ottobre 2015)
- [8W] GOOGLE ANALYTICS
https://www.google.com/intl/it_it/analytics/
(visitato il 17 ottobre 2015)
- [9W] HTML5 API
<http://dev.w3.org/html5/html-author/>
(visitato il 17 ottobre 2015)
- [10W] HTML2CANVAS
<http://www.html2canvas.hertzen.com>
(visitato il 17 ottobre 2015)
- [11W] MOVE.JS
<https://visionmedia.github.io/move.js/>
(visitato il 17 ottobre 2015)
- [12W] FLOT.JS
<https://www.flotcharts.org/>
(visitato il 17 ottobre 2015)

[13W] FONT AWESOME

<https://fontawesome.github.io/Font-Awesome/>

(visitato il 17 ottobre 2015)

[14W] FONTELLO

<https://www.fontello.com/>

(visitato il 17 ottobre 2015)

[15W] HAMMER.JS

[https:// hammerjs.github.io/](https://hammerjs.github.io/)

(visitato il 17 ottobre 2015)

[16W] ALMENDE CHAP LINKS LIBRARY - TIMELINE

<http://almende.github.io/chap-links-library/timeline.html>

(visitato il 17 ottobre 2015)

[17W] TWITTER BOOTSTRAP

<http://www.getbootstrap.com>

(visitato il 17 ottobre 2015)