

eFrame

Relazione di Multimedia e Tecnologie Interattive

Luca Rinaldi, Octavian Bujor

Università di Bologna

Sommario *eFrame è una cornice digitale domestica che mostra informazioni all'utente attraverso uno schermo E-ink. L'autonomia di questo dispositivo è nell'ordine dei mesi e l'utilizzo della filosofia "Calm Technology" permette all'utente di essere informato senza dovere interagire direttamente con esso. Questa relazione mostra la progettazione e lo sviluppo di un prototipo funzionante di eFrame.*

1 Introduzione

In questa relazione illustriamo il progetto eFrame, ovvero una cornice digitale domestica integrata ad uno schermo E-ink per la visualizzazione di informazioni aggregate dal web. L'obiettivo che ci siamo dati per questo progetto è la progettazione e la realizzazione di un prototipo funzionante di tale dispositivo.

Analizzando l'andamento della tecnologia degli ultimi anni ci sono sempre più dispositivi mobili disponibili sul mercato: prima gli smartphone, poi i tablet e recentemente gli smartwatch. Questi dispositivi ci rendono sempre più connessi e sempre più informati su cosa accade nel mondo. Con l'arrivo degli smartphone è nata la necessità di avere sempre più applicazioni installate e successivamente quella di avere più dispositivi "smart". Utilizzando questi strumenti quotidianamente, gli utenti sentono il bisogno di essere sempre aggiornati con le ultime notizie (che siano esse di cronaca, sportive, social, etc.) e un uso eccessivo degli smartphone può portare addirittura a togliere tempo per attività più importanti. Purtroppo questi dispositivi, con l'avanzare della tecnologia, hanno un consumo energetico sempre più alto. L'utente deve impegnarsi a ricaricarli quotidianamente e spesso può capitare di rimanere senza batteria dopo poche ore di utilizzo.

Le aziende cercano di superare il problema dell'autonomia in maniera diversa, alcune offrono una batteria più grande ed altre cercano di progettare dispositivi che necessitano di meno energia possibile. Pebble, ad esempio, ha proposto su Kickstarter il suo progetto *Pebble Time*: uno smartwatch dotato di schermo e-paper a colori con un'autonomia fino a 7 giorni. Il progetto ha raggiunto in pochissimo tempo i 500 mila dollari richiesti ed è arrivato ad ottenere complessivamente più di 20 milioni di dollari di finanziamento. I clienti hanno dimostrato che su un dispositivo del genere la qualità dell'immagine dello schermo non è molto importante se in cambio esso offre un'autonomia decisamente maggiore

rispetto alla concorrenza (ad esempio l'autonomia di Apple Watch arriva fino a 18 ore).

Partendo da queste considerazioni abbiamo pensato di progettare eFrame, un dispositivo domestico particolare e diverso da quelli già esistenti, dotato di batteria ma senza i problemi odierni di autonomia.

1.1 Descrizione

Il dispositivo eFrame eredita tutte le caratteristiche di una cornice classica e la estende attraverso l'utilizzo di uno schermo in grado di visualizzare fino a quattro diverse tipologie di informazioni aggregate dal web nello stesso momento, come ad esempio il meteo o le news. Essendo eFrame dotato di connettività wireless, per il suo utilizzo quotidiano è sufficiente associarlo alla propria rete WiFi domestica.

Grazie alla batteria di lunga durata integrata in eFrame, la cornice può essere posizionata in qualsiasi luogo all'interno della copertura WiFi, anche lontano dalle prese elettriche. Pur potendolo spostare facilmente, eFrame non è stato studiato come dispositivo mobile, in quanto una delle sue caratteristiche principali è la bassa interattività che l'utente deve avere con esso.

Il contenuto visualizzato sullo schermo può essere gestito comodamente dall'utente attraverso l'utilizzo di dispositivi di interazione connessi alla rete, come smartphone o pc. La tipologia del contenuto dipende dalle eApp scelte dall'utente, ovvero delle applicazioni sviluppate apposta per eFrame (come meteo, news, twitter, etc.) in grado di aggregare informazioni anche in base alle preferenze dell'utente, visualizzandole in modo ottimale per facilitare la lettura. Possono essere visualizzate su schermo fino a quattro eApp, in 8 layout differenti (fig. 1). La dimensione delle informazioni di ogni eApp dipende dal layout scelto dall'utente.

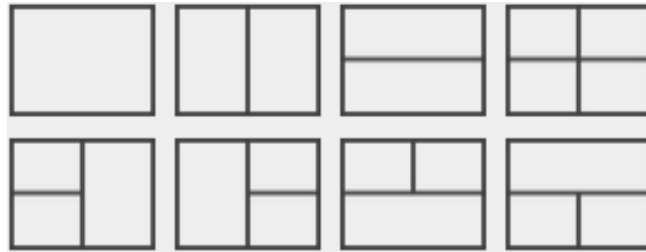


Figura 1. Possibili layout in eFrame

L'utente può anche scegliere la fascia oraria nella quale vuole visualizzare determinate eApp su un determinato layout. In questo modo può personalizzare eFrame in base alle sue abitudini quotidiane. Ad esempio può scegliere di visualizzare le news e il meteo al mattino presto, lasciando il calendario, i tweet e le news per il resto della giornata.

1.2 Caratteristiche principali

Le principali caratteristiche di eFrame sono le seguenti:

Ecosistema cloud L'architettura del sistema è strutturata in modo da garantire il minor carico possibile al dispositivo, elaborando e memorizzando informazioni in cloud. Questo elimina la necessità di eventuali aggiornamenti software derivati dall'aggiunta di nuove applicazioni.

Supporto eApp eFrame offre agli sviluppatori la possibilità di estendere il catalogo delle eApp. Ad esempio l'Università di Bologna potrebbe decidere di sviluppare una semplice applicazione per visualizzare gli orari relativi ai diversi corsi di laurea. In questo caso l'utente potrà visualizzare l'applicazione sul suo eFrame semplicemente selezionandola dal catalogo aggiornato in tempo reale.

Wireless La cornice non ha bisogno di essere costantemente connessa alla rete elettrica e può essere utilizzata senza avere la necessità di essere connessa alla rete internet cablata. Una volta effettuata la configurazione iniziale, la cornice può essere posizionata in qualunque luogo nel raggio del campo WiFi, ad esempio appesa ad un muro o appoggiata su un tavolo.

Efficienza La caratteristica principale della nostra cornice è la grande durata della batteria, per ridurre il carico di gestione affidato all'utente. Pur essendo un dispositivo in grado di prendere informazioni grazie all'utilizzo della rete wireless, la durata della batteria è circa di due mesi.

Schermo E-ink Grazie all'utilizzo della tecnologia e-paper, la frequenza di aggiornamento dello schermo viene ridotta al minimo. Inoltre lo schermo richiede energia solamente durante il cambiamento dell'immagine. In questo modo il dispositivo può spegnersi temporaneamente quando non viene effettuato un aggiornamento.

Bassa interattività Questa caratteristica è la base di eFrame. L'utente deve utilizzare la cornice esattamente come se fosse un quadro. Non è possibile interagire direttamente con la cornice, né tramite lo schermo, né tramite eventuali pulsanti disposti sulla cornice. L'unico modo per configurare la cornice è attraverso i dispositivi di interazione (smartphone, tablet, PC, etc.).

1.3 Calm Technology

Abbiamo scelto di rendere eFrame non interattiva ispirandoci alla teoria Calm Technology (bib. [Calm Technology]).

"I dispositivi dovrebbero lavorare senza interrompere l'attenzione e notificare gentilmente quando è successo qualcosa."

Josh Marinacci

Calm Technology è un tipo di tecnologia dell'informazione in cui l'interazione uomo-macchina avviene nella periferia dell'utente piuttosto che costantemente al centro dell'attenzione. Tra i principi di Calm Technology possiamo trovare:

1. INTRODUZIONE

- La tecnologia utilizzata dovrebbe richiedere la più piccola quantità dell'attenzione dell'utente:
 - La tecnologia può comunicare con l'utente, ma non ha bisogno di parlarci.
 - La conoscenza dell'ambiente dovrebbe essere creata attraverso l'utilizzo di sensi differenti;
 - La tecnologia dovrebbe comunicare informazioni senza distrarre l'utente dal suo compito attuale.
- La tecnologia dovrebbe informare l'utente e calmarsi per il resto del tempo:
 - La tecnologia dovrebbe dare all'utente solo quello che serve per risolvere i suoi problemi, nient'altro.
- La tecnologia dovrebbe fare uso della periferia dell'attenzione:
 - L'utente dovrebbe essere informato senza essere sovraccaricato di informazioni.
- Calm Technology amplifica il meglio della tecnologia e il meglio dell'umanità:
 - Le macchine non dovrebbero comportarsi come umani.
 - Gli umani non dovrebbero comportarsi come macchine.
 - Entrambi i soggetti dovrebbero amplificare le loro parti migliori.

Calm Technology è associata all'*ubiquitous computing* per minimizzare l'invasività percepibile dei dispositivi nella vita di tutti i giorni. eFrame è un buon esempio di ubiquitous computing che segue la teoria di Calm Technology.

"Ubiquitous computing is roughly the opposite of virtual reality. Where virtual reality puts people inside a computer generated world, ubiquitous computing forces the computer to live out here in the world with people."

Mark Weiser
Ricercatore di Xerox PARC

eFrame può essere considerato un dispositivo Calm Technology perché:

1. Non attira l'attenzione dell'utente in modo invasivo: grazie agli aggiornamenti periodici è l'utente che decide di spostare l'attenzione verso la cornice e non è la cornice che spinge l'utente a farlo.
2. La cornice si aggiorna periodicamente senza rendere necessaria l'interazione dell'utente, se non per l'eventuale configurazione attraverso i dispositivi di interazione.
3. Grazie alla durata della batteria e la connessione wireless, l'utente può posizionare ovunque la cornice senza intervenire quotidianamente per la sua accensione, per la sua ricarica o per la sua configurazione.
4. Le informazioni su schermo sono minimali. Ad esempio nella eApp news compare solo il titolo delle notizie e non la notizia intera. In questo modo l'utente non viene sovraccaricato di informazioni ma riceve solo il necessario.

1.4 E-paper

La tecnologia e-paper (bib. [E-ink]), detta anche E-ink, è una tecnologia di display progettata per imitare l'aspetto dell'inchiostro su un foglio di carta. A differenza di un normale schermo, che usa una luce posteriore al display per illuminare i pixel, l'e-paper riflette la luce ambientale come un foglio di carta. Questa tecnologia è stata inventata nel 1996 da Joe Jacobson, fondatore di E-Ink, ed è attualmente la tecnologia più utilizzata per i più diffusi lettori eBook.

La tecnica più utilizzata prevede l'uso di sfere (fig. 2) di dimensione molto ridotta all'interno dello schermo. Queste sfere sono caricate elettricamente; una semisfera è positiva e colorata di nero mentre l'altra semisfera è caricata negativamente e colorata di bianco. Tramite campi elettrici si possono orientare le sfere per ottenere il cambio di colore nei vari punti dello schermo.

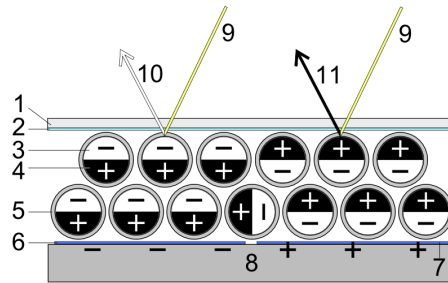


Figura 2. Sfere e-paper

Questo permette di realizzare supporti sottili e che richiedono alimentazione solamente quando si vuole modificare la configurazione delle sfere. Si possono quindi realizzare dispositivi leggeri e a elevata autonomia, dato che l'energia è necessaria solo per cambiare il contenuto dello schermo.

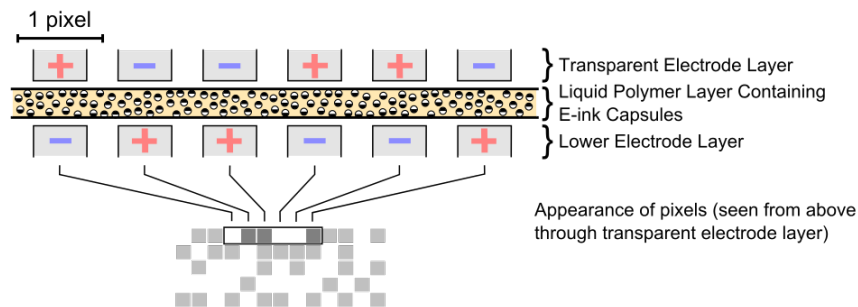


Figura 3. Pixel e-paper

1.5 Dispositivi simili sul mercato

Esistono già dei dispositivi interessanti che sono simili a quello che abbiamo realizzato noi. Molti di questi sono stati progettati per le aziende, quindi per un utilizzo professionale e non per ambienti domestici. I prodotti più comuni sono delle insegne che usano lo schermo E-Ink e offrono connettività WiFi o GSM per aggiornare il contenuto dello schermo. La maggior parte di questi dispositivi permette semplicemente di visualizzare delle immagini caricate in rete e quindi il cliente dovrebbe creare un software per interfacciarsi alle insegne. In questi casi il cliente non potrebbe essere un normale utente inesperto. Inoltre i prezzi sono decisamente alti (minimo \$580) e l'acquisto risulterebbe sveniente per la maggior parte dei clienti privati. I principali progetti che condividono alcuni aspetti con eFrame sono:

PicoSign (bib. [Picosign]) Offre diverse insegne elettroniche dotate di schermo E-ink di diverse dimensioni (da 7" a 10") e di connettività GSM, LAN o WiFi. I dispositivi si aggiornano decine di volte al giorno e hanno una buona autonomia che può arrivare a due anni utilizzando la connessione WiFi. Gli utenti finali possono caricare delle immagini da far visualizzare, gli sviluppatori invece possono usare le WebAPI ufficiali per poter interfacciare altri software alle insegne. Non abbiamo trovato informazioni sul prezzo di questi dispositivi. L'azienda ha comunque pubblicato delle statistiche sull'utilizzo e risultano 480 dispositivi connessi che appartengono a 28 clienti diversi.

Visionect (bib. [Visionect]) Ha sviluppato una tecnologia cloud che permette di visualizzare qualsiasi sito o applicazione web sulle insegne digitali. Le insegne hanno lo schermo E-ink da 6" fino a 13.3" e connettività WiFi o 3G. L'autonomia della batteria risulta molto interessante e può arrivare fino a 3 anni con una carica. Il target di clienti sono le aziende che devono sviluppare una parte del software o integrare il proprio sito web con le insegne. Sono disponibili diversi kit di sviluppo che vanno dai \$580 per la versione a 6" a \$1980 per la versione a 13.3".

Little Printer (bib. [Little Printer]) È un prodotto molto diverso da quelli descritti sopra: si tratta di una piccola stampante connessa al web pensata per la casa. Si possono inviare messaggi o immagini dallo smartphone, oppure abbonarsi a più di 160 pubblicazioni diverse gratuitamente. Le pubblicazioni scelte possono stampate in base alla fascia oraria. Little Printer notifica una nuova stampa disponibile tramite un led e tramite la pressione di un tasto la stampante inizia a stampare. Veniva venduto a un prezzo di \$199, purtroppo da giugno 2015 hanno sospeso i servizi web e quindi hanno smesso di mantenere i dispositivi venduti.

Questa piccola stampante si avvicina molto a quello che ci siamo immaginati noi per eFrame: entrambi sono pensati per un uso domestico e gli utenti non devono preoccuparsi di programmare o di interfacciare il dispositivo con l'utilizzo di API. Anche l'interazione che ha l'utente è simile: in entrambi i casi si personalizza il contenuto tramite un altro dispositivo. Ci sono anche delle differenze: eFrame non ha bisogno di carta e quindi l'utente non deve

preoccuparsi di comprarla. Inoltre, a differenza di Little Printer, eFrame è alimentato a batterie e può essere appeso al muro o spostato comodamente.

2 Progettazione

2.1 Panoramica del sistema

Abbiamo suddiviso l'ecosistema eFrame in tre parti principali: la cornice, il server e i dispositivi di interazione. Uno schema sintetico di questa suddivisione è descritto in figura 4.

Cornice Il ruolo della cornice è minimale: essa, periodicamente (in base ad un timeout definito dal server), deve richiedere al server un aggiornamento e sostituire l'immagine dello schermo con la relativa risposta. La cornice deve solo visualizzare l'immagine generata dal server senza effettuare altre computazioni che ridurrebbero la sua efficienza. In questo modo la cornice opera come *thin client*, ovvero dipende totalmente dal server.

Server centrale Il server contiene tutte le eApp sviluppate ed effettua il rendering dell'immagine per mandarlo alla cornice. Quando riceve una richiesta di aggiornamento da parte della cornice, il server costruisce l'immagine attraverso le impostazioni personali dell'utente (come layout e eApp scelte) ricevute precedentemente dai dispositivi di interazione.

Dispositivi di interazione Sono tutti quei dispositivi personali dotati di browser web o di sistemi operativi mobile (Android o iOS). L'utente può utilizzare la web app (o la mobile app) effettuando il login con i suoi dati personali. Attraverso la web app, l'utente può configurare il suo eFrame selezionando layout, fasce orarie e relative eApp (attraverso un catalogo di eApp costantemente aggiornato). Le informazioni salvate vengono inviate al server e memorizzate su di esso.

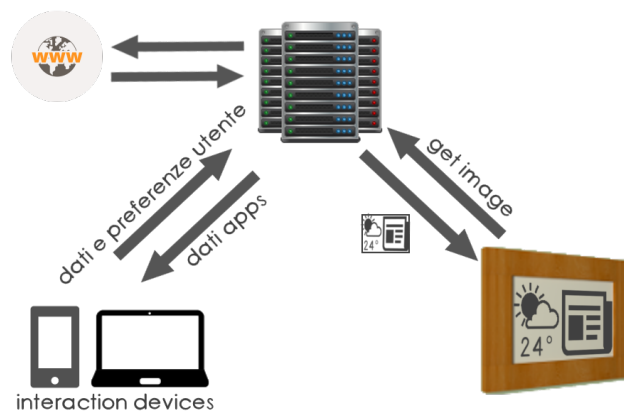


Figura 4. Schema sistema

2.2 Applicazioni

Una applicazione per eFrame (eApp) definisce il contenuto di una porzione dello schermo in base al layout scelto (fig. 1). Le informazioni mostrate da una eApp sono frutto di una aggregazione tra dati utente (preferenze) e dati web.

Ogni eApp può essere visualizzata nello schermo in quattro viste differenti:

- Cella singola;
- Vista orizzontale (due celle adiacenti orizzontalmente);
- Vista verticale (due celle adiacenti verticalmente);
- Schermo intero (quattro celle).

In base alle specifiche della eApp, le viste possono essere visualizzate come un semplice ridimensionamento rispetto a quella a schermo intero, oppure possono differenziarsi aggiungendo più informazioni su viste più grandi.

Una eApp sviluppata correttamente non dovrebbe contenere i seguenti elementi:

Input L'eApp non può ricevere input dall'utente direttamente dal dispositivo. Quindi eventuali elementi di input (come link, pulsanti, form, etc.) sarebbero inutili.

Animazioni e audio A causa del basso intervallo di aggiornamento dello schermo, eventuali animazioni non sarebbero visualizzabili. Inoltre la cornice non può riprodurre audio in quanto non dotata di relative interfacce di output.

Colori Tutte le immagini a colori sono visualizzate tramite una scala di grigi, quindi l'eApp non dovrebbe concentrarsi sull'utilizzo dei colori.

Nel caso l'eApp necessiti di input da parte dell'utente (come le preferenze), può definire una vista dedicata ai dispositivi di interazione. Questa vista rappresenta la pagina di preferenze della eApp mostrata all'utente, in cui può inserire informazioni come ad esempio la sua località, la sua età o i suoi interessi.

2.3 Dispositivi di interazione

I dispositivi di interazione consentono all'utente di configurare eFrame in base alle proprie preferenze. Attraverso questi dispositivi è possibile configurare il layout dello schermo ed associare le eApp da visualizzare ad ogni vista contenuta nel layout. Inoltre è possibile scegliere diverse configurazioni in base a diverse fasce orarie giornaliere. Ogni eApp, una volta selezionata dal catalogo (costantemente aggiornato con le informazioni sul server), può essere configurata in base alle preferenze dell'utente.

Come dispositivi di interazione è possibile utilizzare PC (attraverso browser), smartphone o tablet (attraverso una applicazione dedicata).

3 Sviluppo



Figura 5. Prototipo di eFrame

Non potendo realizzare un dispositivo hardware partendo da zero (e non essendo questo il nostro scopo), per la fase di sviluppo abbiamo deciso di creare un prototipo di eFrame modificando un dispositivo hardware già esistente. Dunque il nostro scopo per la realizzazione di questo progetto è stato quello di creare un prototipo funzionante ad alto risparmio energetico in grado di comunicare con il server. Il dispositivo con schermo eInk che si è più avvicinato alle nostre esigenze è un eBook-Reader con connettività WiFi. Inoltre, per poter modificare facilmente il software, ci siamo orientati verso un dispositivo più aperto, come quelli offerti da Kobo. In particolare abbiamo scelto di utilizzare Kobo Touch.

3.1 Kobo Touch: caratteristiche tecniche

Il modello Touch di Kobo offriva il miglior rapporto qualità/prezzo per le nostre esigenze a scopo dimostrativo. Le sue principali caratteristiche sono:

- Schermo touch screen E Ink Pearl, 6", risoluzione 800x600 e 16 scale di grigio.

3. SVILUPPO

- Connettività Wi-Fi (802.11 b/g/n) e USB 2.0.
- Dimensioni 165 x 114 x 10mm.
- CPU Freescale i.MX508 800 MHz ARM Cortex A8.
- Memoria di massa 2GB.
- Batteria Li-Ion da 1000 mAh.

Una caratteristica fondamentale ai fini del nostro progetto è che il processore presente sul dispositivo consente di sospenderlo con il supporto a *RTCWake*, che permette di svegliarlo dopo un determinato periodo di tempo. Durante la fase di sospensione vengono spenti quasi tutti i componenti hardware dell'eBook-reader, tranne la RAM e una piccola parte della CPU. Grazie alla tecnologia E-ink, anche dopo lo spegnimento dello schermo, il contenuto rimane visualizzato senza subire modifiche.

Per adattarlo al nostro scopo abbiamo dovuto apportare alcune modifiche hardware e software al dispositivo.

3.2 Prototipo: modifiche hardware

Pur essendo adatto alle nostre esigenze, Kobo Touch offre delle caratteristiche che vanno oltre al nostro scopo (come un processore troppo potente o la presenza del touch) e che influiscono negativamente sull'autonomia complessiva del dispositivo. Per garantire un'autonomia di lunga durata, come descritto precedentemente, abbiamo dovuto aggiungere due batterie da 1500 mAh, portando la capacità complessiva a 4000 mAh.

Inoltre, per rafforzare l'aspetto estetico del prototipo e integrarlo nell'ambiente domestico, abbiamo realizzato una cornice di legno su misura per l'eBook-reader. Abbiamo anche inciso a laser il logo di eFrame sulla cornice.

3.3 Prototipo: modifiche software

Il sistema operativo di Kobo Touch è una distribuzione Linux minimale sviluppata dall'azienda con la maggior parte del codice rilasciato open-source, che ci ha consentito di apportare modifiche software al sistema.

L'interfaccia grafica presente precedentemente non risultava utile per il nostro scopo, dunque abbiamo deciso di rimuoverla.

Abbiamo inoltre sviluppato un piccolo modulo software per gestire gli aggiornamenti periodici e per collegare il prototipo al server. Il modulo realizzato (in Python e in Bash) consiste in un ciclo infinito che effettua i seguenti passi:

1. Accensione della scheda di rete WiFi e connessione alla rete domestica;
2. Richiesta al server inviando identificativo del dispositivo e livello della batteria;
3. Attesa della risposta del server;
4. Decodifica della risposta JSON contenente l'immagine (base64 di un PNG convertito e compresso in RAW-rgb565 per l'eBook-reader) e il numero dei secondi prima del prossimo risveglio;

5. Aggiornamento dello schermo con la nuova immagine;
6. Spegnimento della scheda di rete WiFi;
7. Sospensione (tramite RTCWake) per il numero di secondi ricevuti dal server.

Questo modulo viene avviato automaticamente subito dopo il boot del dispositivo.

3.4 Applicativi server

Per realizzare la parte server abbiamo utilizzato diversi strumenti di sviluppo come la piattaforma NodeJS, il motore di templating Handlebars e PhantomJS. Per la memorizzazione delle preferenze relative ai vari dispositivi abbiamo utilizzato MongoDB, ovvero un DMBS NoSQL. Abbiamo scelto di utilizzare le tecnologie web per la loro efficacia nella rappresentazione delle informazioni. Inoltre la generazione dell'immagine finale è resa più semplice dall'utilizzo di applicazioni web a cui si può semplicemente "scattare una foto" tramite PhantomJS. In base alle nostre competenze ci è risultato più semplice utilizzare questo tipo di tecnologie invece che comporre l'immagine disegnando su una canvas.

La parte server è composta da due componenti:

Contenitore eApps Questa componente può essere vista come un CMS (Content Management System) ed è un'applicazione web che si occupa di gestire le eApp. Consente di gestire le richieste della cornice e, in base all'identificativo, utilizza le preferenze dell'utente (salvate precedentemente tramite un dispositivo di interazione) per aggregare le informazioni e disporre le eApp nel layout scelto in base alla fascia oraria attuale.

Convertitore in immagine Questa componente prende in input la pagina web prodotta dal contenitore eApp per inviare alla cornice un'immagine completa da visualizzare sullo schermo. L'immagine viene convertita nel formato RAW-rgb565 e viene poi compressa con gZip per minimizzare il traffico di rete. Il file compresso viene codificato in base64 per poter essere trasmesso nel formato JSON alla cornice, includendo il numero di secondi prima del risveglio.

3.5 Applicazioni eApps

Le eApp sono delle semplici applicazioni web in grado di aggregare dati per restituire un contenuto ottimizzato in base al layout scelto dall'utente. Sono composte da diversi file:

Manifest File JSON che contiene il nome dell'applicazione e la sua descrizione.

Controller File Javascript che si occupa dell'aggregazione dei dati web in base alle eventuali preferenze dell'utente.

Viste File HTML che utilizzano il template Handlebars per costruire una porzione di una pagina web in cui vengono rappresentate le informazioni restituite dal controller, così come devono essere visualizzate sullo schermo. Per ogni vista elencata nel paragrafo 2.2 esiste un file diverso.

Vista per dispositivi di interazione File HTML che utilizza Handlebars per definire una schermata di configurazione della eApp dove l'utente può inserire le sue preferenze.

Risorse Stili (LESS e CSS) o immagini utilizzate dalle viste per rappresentare meglio le informazioni.

Abbiamo deciso di organizzare le eApp in questo modo per rendere molto semplice il loro sviluppo. Utilizzando questo sistema, uno sviluppatore anche non molto esperto può creare la sua eApp in poco tempo. Quindi gli utenti, in futuro, potranno scegliere le eApp da visualizzare da un catalogo aggiornato costantemente dagli sviluppatori di eApps. Abbiamo sviluppato alcune applicazioni di esempio per il prototipo:

- Meteo, con possibilità di configurazione della città (informazioni prese da OpenWeatherMap);
- News, dove è possibile scegliere il genere di notizia da visualizzare (informazioni prese da RSS di Google News);
- Todo list, con la possibilità di inserire le cose da fare e di spuntare quelle già fatte;
- Twitter per la visualizzazione degli ultimi tweet di account popolari in Italia (tramite utilizzo di API Twitter);
- Vignette umoristiche giornaliere di XKCD.

3.6 Dispositivi di interazione



Figura 6. Webapp: login

Per la configurazione di eFrame, l'utente può utilizzare i dispositivi di interazione (PC, smartphone, tablet, etc.). Abbiamo sviluppato una applicazione web

(per PC) e una applicazione Android che permettono all'utente di configurare la propria cornice in base alle proprie preferenze.

Abbiamo progettato l'interfaccia grafica in modo da risultare semplice all'utilizzo anche per gli utenti meno esperti. Le configurazioni salvate sono inviate al server e vengono memorizzate su di esso in base all'identificativo della cornice associata all'utente.

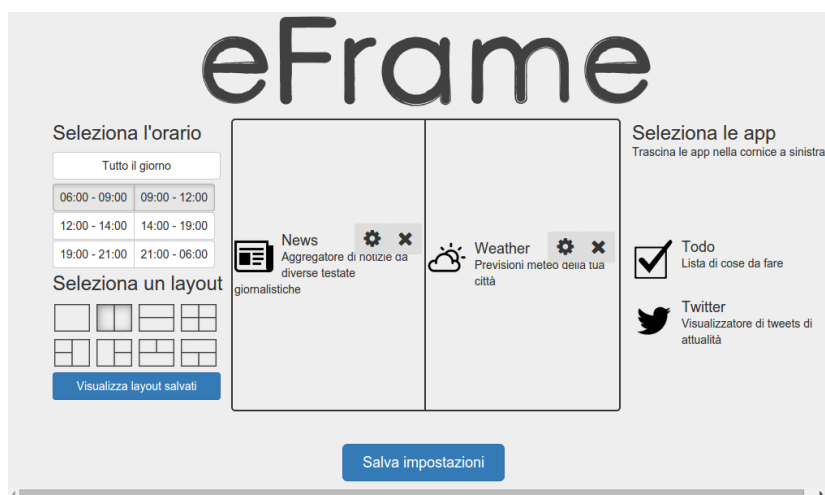


Figura 7. Webapp: configurazione

L'utente può accedere alla configurazione del layout effettuando il login sulla web app (fig. 6). L'immagine 7 mostra la pagina di configurazione di eFrame, in cui l'utente, attraverso i seguenti passi, può personalizzare la visualizzazione delle informazioni sullo schermo:

1. Selezionare la fascia oraria (in alto a sinistra) in cui si vuole visualizzare il layout che si sta configurando. Si possono scegliere più intervalli di tempo, oppure cliccare su "Tutto il giorno";
2. Selezionare il layout scelto (in basso a sinistra) in base a quante eApp si vogliono utilizzare (da 1 a 4). Di solito le eApp assegnate a celle più piccole contengono meno informazioni;
3. Trascinare le eApp scelte (dalla lista a sinistra) nel layout disegnato al centro della pagina. Una volta assegnate, le eApp possono essere configurate o rimosse premendo i relativi pulsanti in alto a destra);
4. Salvare il layout premendo il pulsante "Salva impostazioni" in basso. In questo modo le informazioni salvate vengono inviate al server che si occuperà di memorizzarle per inviare alla cornice la giusta immagine al prossimo aggiornamento.

4. CONCLUSIONE

Inoltre l'utente può visualizzare tutti i layout salvati in base alla fascia oraria (premendo il tasto "Visualizza layout salvati" in basso a sinistra), per poi rimuoverli nel caso voglia cambiare configurazione.

4 Conclusione

Il prototipo di eFrame che abbiamo realizzato soddisfa i nostri obiettivi iniziali e ci è risultato molto comodo all'uso. Dopo averlo posizionato e configurato, i nostri sguardi si dirigevano (spesso involontariamente) anche più volte al giorno verso la cornice, tenendoci informati su cose che molto probabilmente non saremmo andati a cercare manualmente con gli altri dispositivi a nostra disposizione (come ad esempio le informazioni fornite dalle eApp Todo-list e News).

Abbiamo configurato il prototipo per aggiornarsi automaticamente ogni 30 minuti durante il giorno (06:00-24:00) e ogni 2 ore durante la notte, per un totale complessivo di 39 aggiornamenti ogni 24 ore. Assumendo che ogni aggiornamento dello schermo abbia la durata di circa 20-30 secondi e osservando il decremento giornaliero della percentuale della batteria, abbiamo stimato una autonomia complessiva di circa 2 mesi (tra le 6 e le 8 ricariche all'anno). Considerando che si tratta solo di un prototipo, pensiamo che l'autonomia possa essere incrementata dopo aver fatto alcune modifiche hardware proposte nel paragrafo 4.2.

Nei prossimi due paragrafi illustriamo le nostre idee principali per eventuali sviluppi futuri di eFrame.

4.1 Possibili eApp future

Oltre alle cinque eApp già presenti nel nostro prototipo, abbiamo pensato anche ad altre possibili aggiunte che potrebbero risultare interessanti:

- Orari universitari. L'utente potrebbe configurare l'eApp inserendo il codice e l'anno accademico del proprio corso.
- Calendario. L'utente potrebbe configurare l'eApp collegandola al proprio Google Calendar. Potrebbe posizionare eFrame sulla scrivania o sul muro per essere sempre informato sugli eventi futuri.
- Blocco note. L'utente potrebbe inserire dei promemoria per lui o per la sua famiglia.
- Quadro. Questa eApp andrebbe utilizzata solo con il layout a schermo intero e l'utente potrebbe inserire l'immagine (meglio se in bianco e nero) da visualizzare.

4.2 Problemi e sviluppi futuri

Il prototipo che abbiamo realizzato è funzionante ed ha una buona autonomia ma ovviamente il suo hardware (Kobo Touch) non è nato per questo scopo. Il processore, ad esempio, ci è risultato troppo potente per le operazioni che la cornice deve fare e le funzionalità touch non vengono utilizzate. Questi elementi

di consumo potrebbero benissimo essere limitati o rimossi in un eventuale dispositivo in commercio. Pensiamo che sviluppando un dispositivo hardware ad hoc che limiti i consumi di eFrame, sia possibile raddoppiare (se non triplicare) la sua autonomia.

Abbiamo anche pensato alla configurazione iniziale di eFrame, subito dopo averlo acceso per la prima volta: sul retro della cornice (o in un foglio compreso nella confezione) ci potrebbero essere i dati di login per accedere alla piattaforma web sui dispositivi di interazione. Il problema che rimane da risolvere è la configurazione per l'accesso alla rete WiFi domestica, ecco le nostre idee:

- La cornice potrebbe fare da Access Point per consentire all'utente di connettersi direttamente ad una pagina di configurazione interna. Naturalmente servirebbe una scheda di rete adatta e, per non influire negativamente sull'autonomia, questa funzionalità andrebbe accesa solo quando la cornice non riesce a connettersi alla rete.
- Utilizzando una porta USB si potrebbe collegare un dispositivo alla cornice per le configurazioni manuali.
- Si potrebbe dotare la cornice di un ingresso ethernet. A differenza dei punti precedenti, questa soluzione garantirebbe una maggiore autonomia e permetterebbe alla cornice di connettersi alla rete anche in assenza di connessioni WiFi.

Come ultimo punto abbiamo pensato anche alle possibili evoluzioni di eFrame fuori dall'ambiente domestico:

- I commercianti potrebbero utilizzare la cornice per consentire ai clienti di visualizzare gli orari di apertura e chiusura del proprio negozio, in base al giorno. Se pensiamo ad un solo aggiornamento giornaliero (e non ogni 30 minuti) l'autonomia della batteria avrebbe un grandissimo aumento.
- La cornice potrebbe essere utilizzata nelle scuole o nelle università per mostrare gli orari o come bacheca degli annunci. In questo secondo caso ci vorrebbe uno schermo abbastanza grande (come quelli sviluppati da E-ink per la segnaletica stradale) e si potrebbe aggiungere la possibilità di inserire annunci direttamente tramite NFC (sarebbe possibile inserire o togliere annuncio semplicemente toccando il dispositivo con uno smartphone). Vogliamo fare notare che con questa soluzione molte delle caratteristiche principali di eFrame verrebbero a mancare: ci sarebbe più interattività e l'autonomia sarebbe sicuramente ridotta al minimo (per cui la bacheca dovrebbe essere collegata alla rete elettrica).
- I ristoranti potrebbero utilizzare una versione della cornice come menù, sia esterno (con uno schermo più grande) che interno (più piccola e maneggevole). Con un menù di una pagina il dispositivo potrebbe rimanere spento per la maggior parte del tempo e quindi la sua autonomia sarebbe sicuramente maggiore. Nel caso di menù a più pagine si potrebbero inserire dei pulsanti o il touch, ma diventerebbe molto simile ad un normale E-book reader.

Riferimenti bibliografici

[Calm Technology] <http://calmtechnology.com/>
[E-ink] https://it.wikipedia.org/wiki/Carta_elettronica
[Picosign] <http://www.picosign.com/>
[Visionect] <https://www.visionect.com/>
[Little Printer] <http://littleprinter.com/>