

Sound fix

del Puente Guido, Festa Biagio, Montanile Giovanni

Abstract

<<<<<<<<<<<<<<<<<Abstract, should normally be not longer than
200 words.Abstract, should normally be not longer than 200 words.Abstract,
should normally be not longer than 200 words.Abstract, should normally be
not longer than 200 words.Abstract, should normally be not longer than 200
words.Abstract, should normally be not longer than 200 words.Abstract, should
normally be not longer than 200 words.Abstract, should normally be not longer
than 200 words.

1. Tag ID3

I file audio digitali possono contenere, oltre alla traccia audio, testo e/o informazioni grafiche. Le informazioni possono essere per esempio il titolo del brano, il nome dell'artista, il nome dell'album, anno e genere. Queste sono visualizzate quando viene riprodotto un file audio digitale sul computer o dispositivo portatile. Il processo di includere informazioni diverse da quelle del suono nei file audio digitali è comunemente indicato come "tagging". Attraverso questo processo il file audio viene arricchito con le informazioni aggiuntive che descrivono il file audio i "tag". Lo standard originale per file digitali di tagging è stato sviluppato nel 1996 da Eric Kemp che ha coniato il termine ID3. A quel tempo ID3 semplicemente significava "identificare un MP3". I Tag ID3 sono stati progettati per il formato MP3. Con ID3v2 sono diventati disponibili per i formati container-based come Ogg Vorbis e WMA.

ID3v2 informal standard [ID3v2]



ID3v2 è stato progettato per essere il più flessibile ed espandibile possibile per soddisfare le nuove esigenze di meta-informazioni che potrebbero sorgere. Per raggiungere questo scopo ID3v2 è costruito come un contenitore per diversi blocchi di informazioni, cosiddetti frames, il cui formato non è necessariamente noto al software. All'inizio di ogni frame c'è un unico identificatore predefinito, un

descrittore di dimensione che consente al software di saltare frames sconosciuti, e un campo di flags. I flags descrivono i dettagli relativi alla codifica.

Struttura complessiva del Tag:

Header
Extended Header
Frames
Padding
Footer

in generale padding e footer sono mutualmente esclusivi

- ID3v2 header

La prima parte del tag ID3v2 è costituita dall'header formato da 10 byte, disposti come segue: ID3v2/file identifier "ID3" ID3v2 version ID3v2 flags ID3v2 size I primi tre byte del tag sono sempre "ID3", per indicare che è un tag ID3v2, seguiti dai due byte che indicano la versione.

- ID3v2 footer

Per accelerare il processo di individuazione di un tag ID3v2 durante la ricerca dalla fine di un file, può essere aggiunto al tag un footer. Il footer è una copia dell'intestazione, ma con un identificatore differente. ID3v2 identifier "3DI" ID3v2 version ID3v2 flags ID3v2 size

- ID3v2 frame

Tutti i frame ID3v2 sono costituiti da un frame header seguito da uno o più campi contenenti le informazioni effettive. L'intestazione è sempre 10 byte e suddivisa come segue: Frame ID Size Flags. Non c'è un ordine fisso dei frame nel tag. Un tag deve contenere almeno un frame. Un frame deve essere di almeno 1 byte, escludendo l'intestazione.

posizionamento dei Tag

La posizione predefinita di un tag ID3v2 è di precedere l' audio in modo che i player multimediali possono beneficiare delle informazioni quando i dati sono in streaming. E 'comunque possibile accodare il tag, o fare un combinazione delle due.

<nome libreria usata>

noi abbiamo scelto di usare la libreria <nome libreria usata> per l'accesso e l'inserimento dei Tag...

2. Fingerprint

<<<<<<<<<<<<<<<audio digitali possono contenere, oltre alla traccia audio, testo e/o informazioni grafiche. Le informazioni possono essere per

esempio il titolo del brano, il nome dell'artista, il nome dell'album, anno e genere. Queste sono visualizzate quando viene riprodotto un file audio digitale sul computer o dispositivo portatile. Il processo di includere informazioni diverse da quelle del suono nei file audio digitali è comunemente indicato come "tagging". Attraverso questo processo il file audio viene arricchito con le informazioni aggiuntive che descrivono il file audio i "tag". Lo standard originale per file digitali di tagging è stato sviluppato nel 1996 da Eric Kemp che ha coniato il termine ID3. A quel tempo ID3 semplicemente significava "identificare un MP3". I Tag ID3 sono stati progettati per il formato MP3. Con ID3v2 sono diventati disponibili per i formati container-based come Ogg Vorbis e WMA. audio digitali possono contenere, oltre alla traccia audio, testo e/o informazioni grafiche. Le informazioni possono essere per esempio il titolo del brano, il nome dell'artista, il nome dell'album, anno e genere. Queste sono visualizzate quando viene riprodotto un file audio digitale sul computer o dispositivo portatile. Il processo di includere informazioni diverse da quelle del suono nei file audio digitali è comunemente indicato come "tagging". Attraverso questo processo il file audio viene arricchito con le informazioni aggiuntive che descrivono il file audio i "tag". Lo standard originale per file digitali di tagging è stato sviluppato nel 1996 da Eric Kemp che ha coniato il termine ID3. A quel tempo ID3 semplicemente significava "identificare un MP3". I Tag ID3 sono stati progettati per il formato MP3. Con ID3v2 sono diventati disponibili per i formati container-based come Ogg Vorbis e WMA. audio digitali possono contenere, oltre alla traccia audio, testo e/o informazioni grafiche. Le informazioni possono essere per esempio il titolo del brano, il nome dell'artista, il nome dell'album, anno e genere. Queste sono visualizzate quando viene riprodotto un file audio digitale sul computer o dispositivo portatile. Il processo di includere informazioni diverse da quelle del suono nei file audio digitali è comunemente indicato come "tagging". Attraverso questo processo il file audio viene arricchito con le informazioni aggiuntive che descrivono il file audio i "tag". Lo standard originale per file digitali di tagging è stato sviluppato nel 1996 da Eric Kemp che ha coniato il termine ID3. A quel tempo ID3 semplicemente significava "identificare un MP3". I Tag ID3 sono stati progettati per il formato MP3. Con ID3v2 sono diventati disponibili per i formati container-based come Ogg Vorbis e WMA.

3. Android

I brand che costruiscono smartphone sono molti. Il presente progetto è completamente open-source, perciò si è scelto di utilizzare il sistema operativo Google Android.

Storia

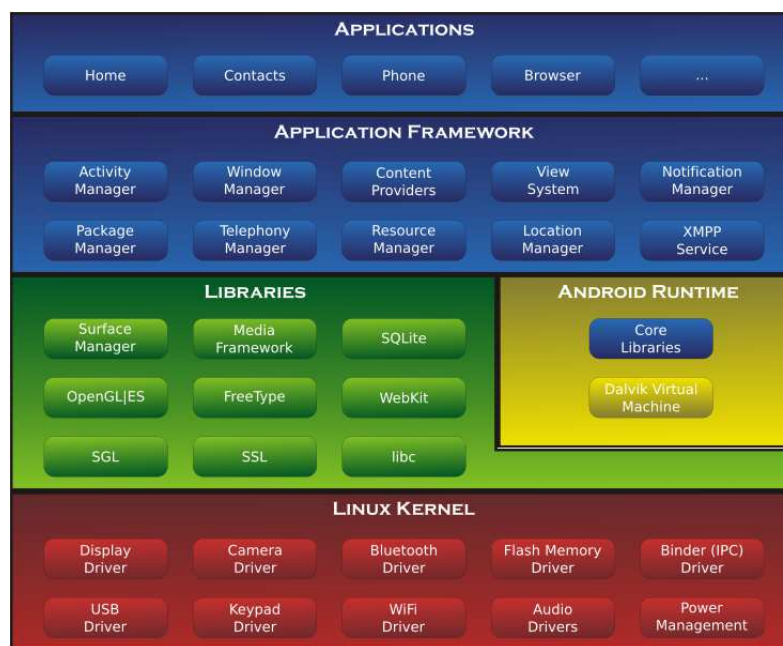
Il primo dispositivo mobile dotato della piattaforma Android è il T-Mobile G1, prodotto dalla società taiwanese HTC. Il prodotto è stato presentato 2008. Dal 2008 gli aggiornamenti di Android per migliorarne le prestazioni e per eliminare

i bug delle precedenti versioni sono stati molti. Allo stato dell'arte si è giunti alla versione 4.2 denominata JellyBean (è stata annunciata la 4.3 all'ultimo Google IO). Per questo progetto abbiamo scelto per una maggiore diffusione di utilizzare la versione 2.3 Gingerbread. Questa versione è stabile e installata sulla maggior parte dei dispositivi ad oggi (questo a causa delle case produttrici che non rilasciano aggiornamenti all'ultima versione).

Abstract

Android è una piattaforma open-source per smartphone e tablet, basata sul sistema operativo Linux. La piattaforma ha a disposizione delle librerie dedicate, un application framework, la Dalvik virtual machine (una Java virtual machine modificata) come ambiente di runtime e una serie di applicazioni preinstallate come un browser, una rubrica e un calendario. Google fornisce agli sviluppatori il software development kit (SDK) che include: strumenti di sviluppo, librerie, un emulatore del dispositivo, la documentazione, alcuni progetti di esempio, tutorial, FAQ, e altro. Per questo progetto abbiamo utilizzato il plug-in per Eclipse(ADT).

Achitettura



L'architettura di Android si presenta suddivisa in diversi livelli, come mostrato in figura. Alla base dello stack Android troviamo un kernel Linux. La scelta di una simile configurazione è nata dalla necessità di disporre di un vero e proprio sistema operativo che fornisca gli strumenti di basso livello per la virtualizzazione dell'hardware sottostante attraverso l'utilizzo di diversi driver.

Le applicazioni Android vengono sviluppate in Java. Serve pertanto una Macchina Virtuale in grado di eseguire le applicazioni. È presente infatti una Macchina Virtuale ottimizzata dalla Sun Microsystem (la Dalvik Virtual Machine) per l'esecuzione di software su sistemi con risorse limitate, quali possono essere ad esempio i dispositivi portatili come gli smartphone. Anche la Dalvik Virtual Machine, similmente per quanto accade per la JVM, implementa un Garbage Collector, liberando pertanto lo sviluppatore dall'onere della gestione della memoria.

Per quanto riguarda, infine, l'Application Framework, lo sviluppatore ha a disposizione una serie di API che sfruttano le librerie sottostanti di Android.

Componenti di una applicazione Android Una caratteristica innovativa del modello di Android, che si differenzia da altre piattaforme, è un supporto per il riutilizzo dei componenti. Con componente si intende un qualsiasi programma, applicazione o servizio implementati. Le applicazioni non sono più pensate come entità a sé stanti, ma come un insieme di componenti disponibili nel sistema e richiamabili all'occorrenza, nel momento in cui servono. Ne scaturisce un vantaggio non indifferente per lo sviluppatore, che si traduce in una maggiore velocità nella realizzazione di una applicazione in quanto non serve reimplementare dall'inizio.

In una applicazione Android esistono più tipologie di componenti. Ne osserviamo due principali:

- **Activities:** una Activity è una entità alla quale è collegata una particolare interfaccia utente. Ogni applicazione per l'interazione con l'utente mostra una serie di activity attraverso le quali è possibile interagire con il sistema sottostante. All'avvio Android esegue l'Activity marcata come principale e da questa, in seguito alle azioni intraprese dall'utente, esegue le successive Activity formando così l'intera interfaccia utente;
- **Services:** sono dei task che vengono eseguiti in background per un indefinito periodo di tempo. Sono ad esempio i processi che lavorano sotto all'interfaccia grafica per elaborare le informazioni inserite dall'utente.

Ciclo di vita dei componenti Tutti i componenti di un'applicazione hanno un ciclo di vita che ha inizio nel momento in cui Android le istanzia con lo scopo di rispondere ad un intent e termina quando le loro istanze vengono distrutte.

- **Activity**

Il ciclo di vita di una Activity, ad esempio, è indicato in figura. Come è possibile notare, una Activity si trova in uno degli stati indicati in figura. Per sua natura Android tenta di mantenere attive più applicazioni possibile, per minimizzare il dispendio di energia per riattivare da zero una applicazione in caso di necessità. È compito del Garbage Collector scegliere quale applicazione rimuovere dalla memoria in caso di necessità.

I tre stati principali in cui si può trovare un Activity sono:

Running: l'activity è visualizzata sullo schermo e l'utente interagisce con essa.

Paused: per quanto l'activity sia visibile l'utente non vi può interagire (ad esempio se abbiamo una finestra di dialogo sovrapposta). Le activities in questa condizione conservano le informazioni riguardanti il loro stato. • Stopped: quando l'utente passa ad un'altra activity, la prima è posta in stato di stop. In questo modo una nuova activity è posta sulla cima della pila. Le activities poste in stop sono terminate prima in caso di esigenza di risorse aggiuntive.

- Service

Un servizio può essere lanciato da un client invocando il metodo *Context.startService(Intent)* e fermato utilizzando il metodo *Context.stopService(Intent)*

Il software viene solitamente distribuito sotto forma di pacchetto autoinstallante, quindi un file con estensione .APK . Questo non è altro che un file compresso, contenente il software (file con estensione .dex) le sue risorse (immagini, suoni ecc...) e alcuni file XML. L'utente medio non ha necessariamente bisogno di conoscere tale tipologia di file, dato che il dispositivo gestisce tutta la parte di installazione mediante web services come Google Play.

Linguaggio

Tramite l'SDK possiamo passare dalla descrizione della nostra applicazione alla sua effettiva esecuzione sia in emulazione, sia su un dispositivo concreto. Per descrivere l'applicazione al dispositivo prescelto si utilizza il file Manifest.xml Possiamo quindi affermare che un'applicazione è descritta completamente da una tripletta:

- Codice Java
- Risorse statiche xml
- Manifest.xml

Java

Il linguaggio per applicazioni Android è un dialetto del linguaggio Java. Anche la virtual machine è diversa (Dalvik). Nella tipica applicazione Android non c'è un entry point (metodo "main") tutto è pensato per essere un "componente" pilotato dagli eventi ("Event Driven").

Risorse XML

Parte dichiarativa contenente varie informazioni: informazioni di layout supporto multilingue

Android Manifest XML

File che descrive l'applicazione al dispositivo. Il Manifest elenca la lista delle necessità del programma per poter operare nel sistema; per esempio, se una apk richiede la connessione alla rete, lo notifica nel Manifest.

4. Gracenote



Pioniera nel settore dei digitalmedia, Gracenote combina informazione, tecnologia, servizi e applicazioni per creare soluzioni di intrattenimento ingegnose per il mercato globale. Dalla gestione dei media, di arricchimento e di tecnologie di identificazione dei contenuti, Gracenote consente ai fornitori di prodotti multimediali digitali di rendere le loro offerte più potenti e intuitive, permettendo esperienze di utilizzo superiori. Le soluzioni dell'azienda integrano la più ampia, più profonda, e la più alta qualità dei metadati e contenuti globali arricchita da una infrastruttura di servizi di miliardi di ricerche al mese da migliaia di prodotti utilizzati da centinaia di milioni di consumatori. I clienti di Gracenote includono i più grandi nomi dell'elettronica, telefonia mobile, automobilistico, software e Internet industrie di consumo. I soci della comunità di intrattenimento includono i principali editori musicali e le etichette indipendenti, prominenti e studi cinematografici.

Tecnologie di Gracenote vengono utilizzate dai principali servizi di media online come Apple iTunes, Spotify e Sony Music Unlimited, per applicazioni mobili innovative come Soundtracking e Path, e dai principali produttori di elettronica di consumo come Sony, Philips e Acer.

Gracenote è il top come Database Multimediale di metadati per questo abbiamo scelto di utilizzarlo nel nostro progetto.

5. Github

GitHub

GitHub è un servizio di hosting web-based per i progetti di sviluppo software che utilizzano il sistema di controllo di revisione Git. GitHub è il più popolare sito di repository di codice open-source. Il sito mette a disposizione funzionalità di social networking, come i feed, i followers e il grafico di rete per visualizzare come gli sviluppatori lavorano su loro versioni di un repository.

Un ottimo servizio che facilita una buona organizzazione del team con molte funzionalità. creazione di Issue, Milestones, revisione del codice, commenti ai commit, sfogliare e modificare on-line il codice sorgente conservando uno storico.

Abbiamo utilizzato il plug in per Eclipse Egit per una migliore fruizione e gestione del progetto condiviso

6. Eclipse



Eclipse è un ambiente di sviluppo integrato multi-linguaggio. La piattaforma di sviluppo è incentrata sull'uso di plug-in, delle componenti software ideate per uno specifico scopo. Inoltre grazie alla sua licenza open source chiunque può sviluppare e modificare i vari plug-in apportando così migliorie e nuove caratteristiche ad una piattaforma già notevolmente avanzata. Nella versione base è possibile programmare in Java, usufruendo di comode funzioni di aiuto quali: completamento automatico, suggerimento dei tipi di parametri o dei metodi, possibilità di accesso diretto a CVS e riscrittura automatica del codice in caso di cambiamenti nelle classi. Nel progetto, Eclipse è stato utilizzato come software principale per lo sviluppo del codice.

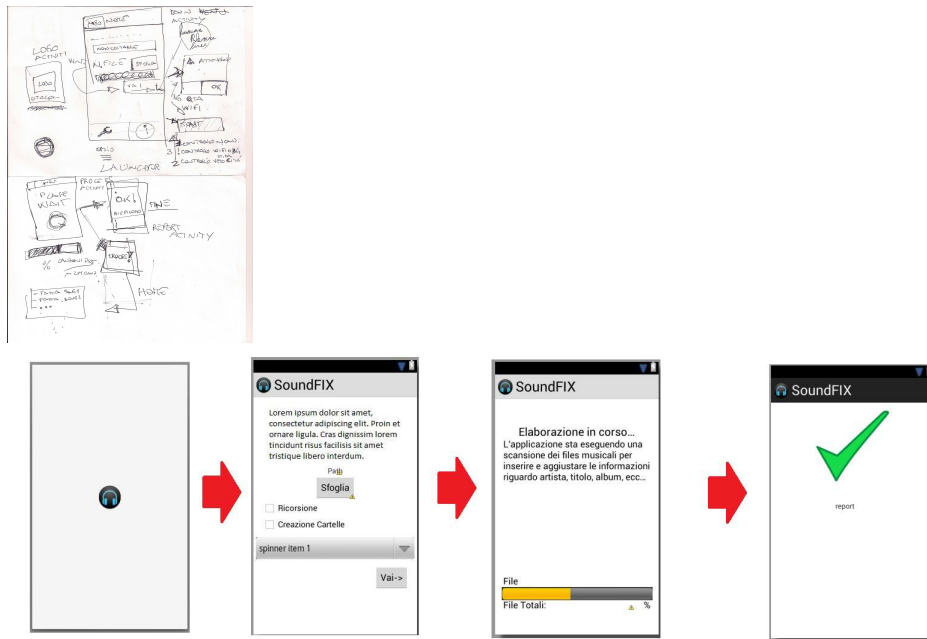
7. Sound fix

Il progetto nasce dall'osservazione di un'esigenza comune degli utenti, di dispositivi mobili. I Player multimediali sono tantissimi e offrono svariate funzionalità, hanno la capacità di presentare la libreria musicale, in possesso dell'utente, in forme dal design intuitivo e attraente. Ma tutte partono da un presupposto "la libreria musicale è fornita di metadati". Questo non sempre è verificato anzi poche volte l'utente si impegna a tenere ordine, poichè questo porta via tempo e non sempre si riesce a venire a conoscenza di tutte le informazioni relative ad una canzone.

Per questo c'è bisogno di Sound fix!

Sound fix non è un'idea innovativa ma è creata per essere vicina all'utente. Molte applicazioni aiutano nell'inserimento di metadati nei file multimediali ma tutte hanno bisogno che l'utente inserisca manualmente o confermi uno ad uno i campi. A differenza dei competitors Sound fix non demanda alcuna fatica all'utente ma automaticamente attraverso metadati già presenti o riconoscendo la canzone dall'impronta audio inserisce tutti i tag nella libreria musicale.

Progetto La progettazione di un'applicazione Android nasce dalla definizione delle Activity di cui si ha bisogno. Abbiamo fissato gli elementi fondamentali per l'utilizzo dell'app. Come si può notare dalle immagini, riportate sotto, sono due le Schermate principali un MainActivity per la presentazione e definizione delle opzioni e una ProcessActivity per l'elaborazione e relativa attesa del processo.



MainActivity

La MainActivity è la schermata iniziale, di partenza dell'applicazione. Nella prima versione del progetto la MainActivity presentava l'applicazione proponendo la scelta delle opzioni e infine l'avvio dell'Activity di elaborazione. Successivamente si è pensato di rendere il tutto più intuitivo e non spaventare l'utente inesperto con i troppi campi. Si è deciso quindi di separare le due cose e inserire le opzioni in un'activity dedicata accessibile dalla Main attraverso il menu a comparsa del tasto opzioni.

Opzioni le opzioni comprendono:

1. cartelle/sottocartelle
 2. Modalità
 3. creazione cartelle
 4. download copertine album
 5. Seleziona cartella
- 1 cartelle/sottocartelle è una checkbox che serve per dare la possibilità di scegliere se attivare o meno la ricorsione di cartella in sottocartella per la ricerca dei file multimediali. Di default è attiva.
 - 3 creazione cartelle è una checkbox che serve ad attivare l'organizzazione della libreria musicale creando una cartella per ogni artista e dentro ad ognuna separare le canzoni per album. Può essere una funzionalità utile ma non è indispensabile per un utente inesperto. Per questo di default è disattivata.

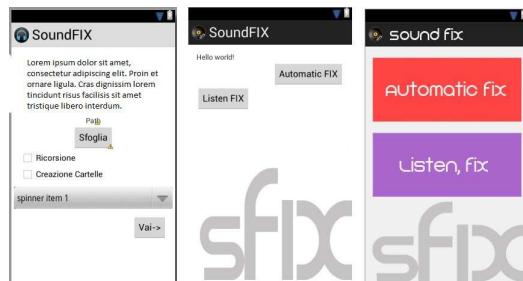
controlla se per ogni file sono presenti tutti i tag se non sono presenti controlla se è presente almeno il titolo se c'è manda una query in formato xml per il riconoscimento degli altri tag se non c'è salta il brano

(default) controlla se per ogni file sono presenti tutti i tag se non sono presenti controlla se è presente almeno il titolo se c'è manda una query in formato xml per il riconoscimento degli altri tag se non c'è esegue l'operazione di fingerprint sulla canzone per il riconoscimento dei tag

controlla se per ogni file sono presenti tutti i tag se non sono presenti esegue l'operazione di fingerprint sulla canzone per il riconoscimento dei tag

<<<<<<<<<<<<<<<<<<<<<La MainActivity è la schermata iniziale, di partenza dell'applicazione. Nella prima versione del progetto la MainActivity presentava l'applicazione proponendo la scelta delle opzioni e infine l'avvio dell'Activity di elaborazione. Successivamente si è pensato di rendere il tutto più intuitivo e non spaventare l'utente inesperto con i troppi campi. Si è deciso quindi di separare le due cose e inserire le opzioni in un'activity dedicata accessibile dalla Main attraverso il menu a comparsa del tasto opzioni.

Design



Il successo di un'applicazione, sul piano progettuale e realizzativo, si basa spesso su un mix sapiente ed equilibrato di tecniche di programmazione potenti ed efficaci, con interfacce eleganti, intuitive, che possano sfruttare al meglio la semplicità di interazione. Il design della User Interface (UI) deve essere considerato un aspetto fondamentale e imprescindibile, perché l'esecuzione curata in ogni dettaglio incide su un miglior utilizzo delle informazioni, sulla velocità e sulla qualità di interazione. La sottovalutazione dell'importanza della Graphics User Interface di un'applicazione si evidenzia spesso quando il progetto è influenzato da scelte esclusivamente tecniche, che impattano solo lo sviluppo software. Google inoltre delinea, nel sito degli sviluppatori, anche delle linee guida per il design di un app Android.

Sono tre i principi fondamentali:

Enchant me (incantami):

Le applicazioni Android sono eleganti ed esteticamente piacevoli.

Simplify my life (semplifica la mia vita):

Le applicazioni Android rendono la vita più facile e sono facili da capire.

Make me amazing (Rendimi incredibile):

Applicazioni Android permettono alle persone di provare cose nuove.

Abbiamo cercato di mantenere la UI più semplice possibile utilizzando icone e colori consigliati da Google per facilitare l'usabilità. Non abbiamo eliminato le opzioni più impegnative ma le abbiamo rese disponibili attraverso un apposito menu. Il tutto arricchito da particolari utili e dal design convincente. Dall'immagine si può notare l'evoluzione che ha avuto l'applicazione dal punto di vista grafico, che ha implicato cambiamenti progettuali e di codice.

Futuri sviluppi

L'applicazione ha percorso due del processo di sviluppo. Il lavoro di perfezionamento non finisce qui ma continuerà con nuovi test e inserimento di nuove funzionalità. Già nell'ultima versione è stata inserita nel progetto una nuova funzionalità. Un'alternativa al fix automatico, "Listen, fix" una modalità che

<<<<<<<Appendix, only when needed. ppendix, only when needed.ppendix,
only when needed.ppendix, only when needed.ppendix, only when needed. ppenn-
dix, only when needed ppendix, only when needed.ppendix, only when need-
ed.ppendix, only when needed.ppendix, only when needed.ppendix, only when
needed.ppendix, only when needed.ppendix, only when needed.ppendix, only
when needed.ppendix, only when needed.ppendix, only when needed.ppendix,
only when needed.ppendix, only when needed.

Or plain bibliography: