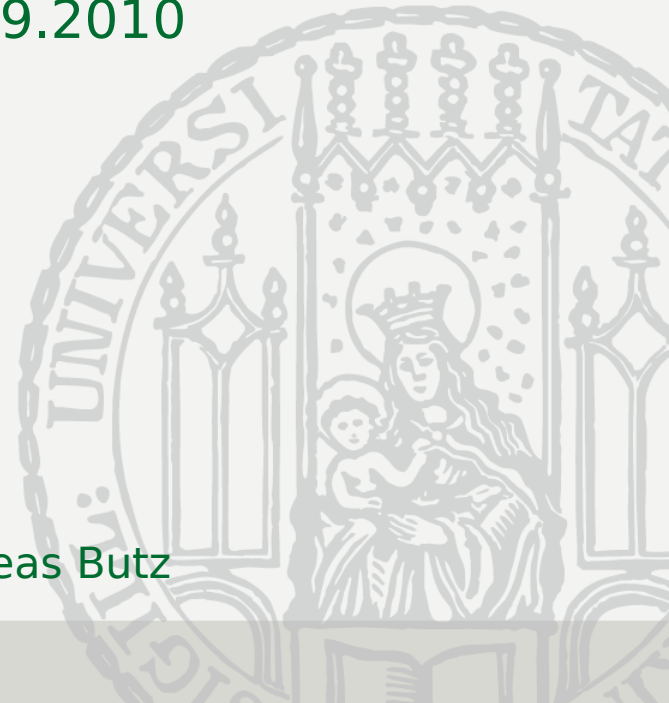


Visualizing multiple Last.fm listening histories

Abschlussvortrag Bachelorarbeit - 14.09.2010

Roman Graebisch
www.lastloop.de

LFE Medieninformatik
Betreuer: Dominikus Baur
Verantwortlicher Hochschullehrer: Prof. Dr. Andreas Butz





Inhalt

- Motivation & Related Work
- LastLoop
- Evaluation
- Zusammenfassung & Ausblick



Motivation



Last.fm – Music Listening Histories

- ermöglicht individuelles loggen von Musiktiteln
- Datensammlungen einfach zugänglich über Benutzername
- Pro Lied: Timestamp, Titel, Interpret, Album; Tags
- Große Datenmengen: oft > 100.000 Einträge/Benutzer



- keine Visualisierung für mehrere Benutzer (> 2)
- Aufgabenstellung: Visualisierung mehrerer Datensätze gleichzeitig
- neue Insights durch Vergleiche zwischen mehreren Nutzern
 - z.B.:
 - gleiche Lieblings- Genre/Band/Lied
 - gegenseitige Beeinflussung

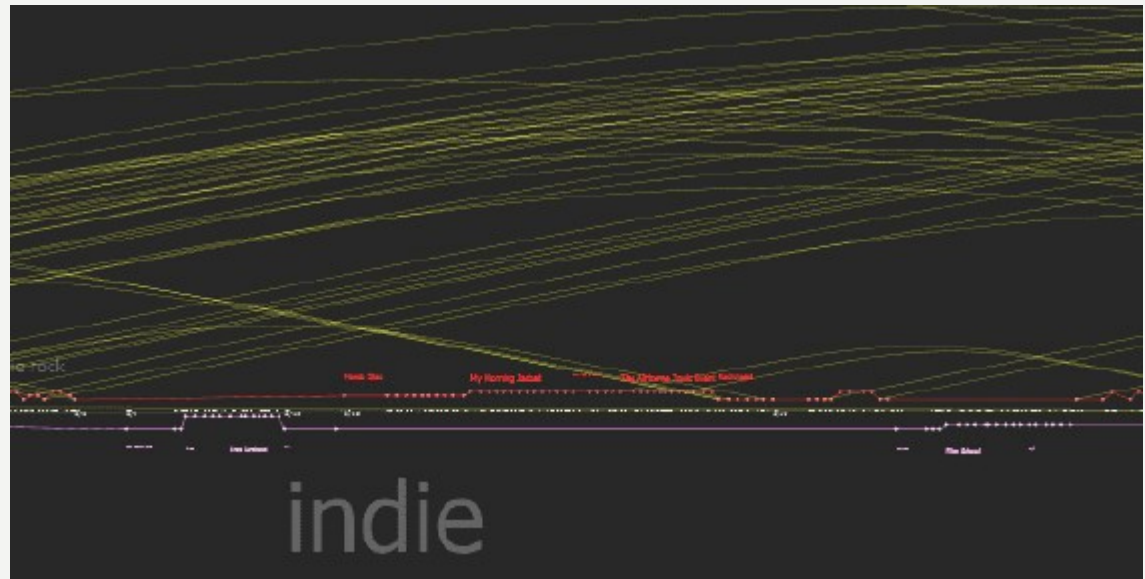


Related Work



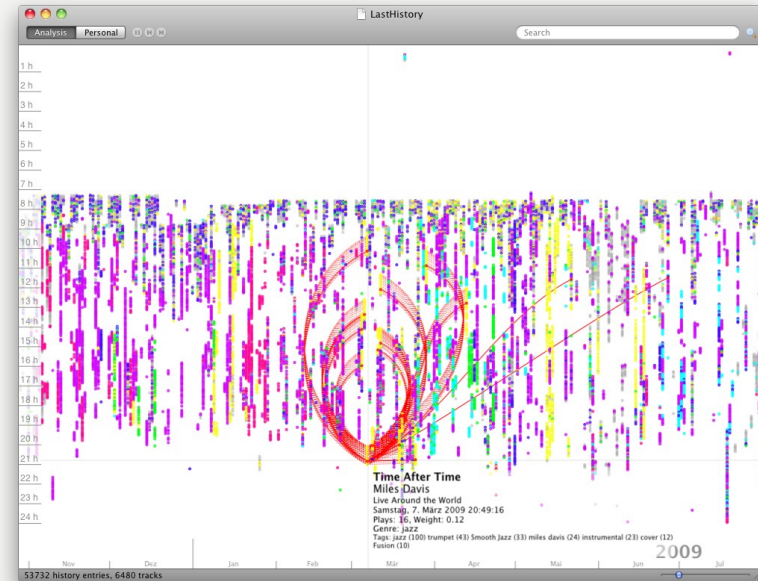
LoomFM⁽¹⁾

- vergleich von 2 Benutzern
- verbinden gleicher Lieder



LastHistory⁽³⁾

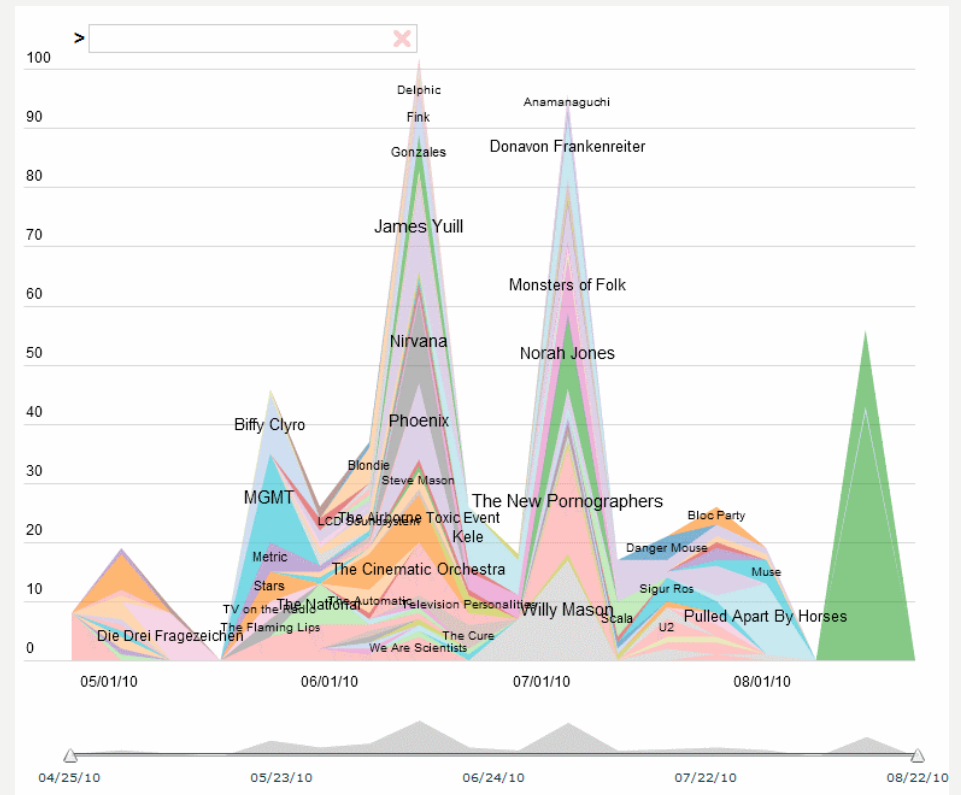
- umfangreiche Analyse einer einzelnen History
- Aufbereitung entlang Zeitachse, Genre und Titel
- erlaubt Mustererkennung
- Personal Mode - Reminiszieren





LastExplorer⁽⁴⁾

- Analyse eines Benutzers
- Stacked Graphs
- Einträge wöchentlich



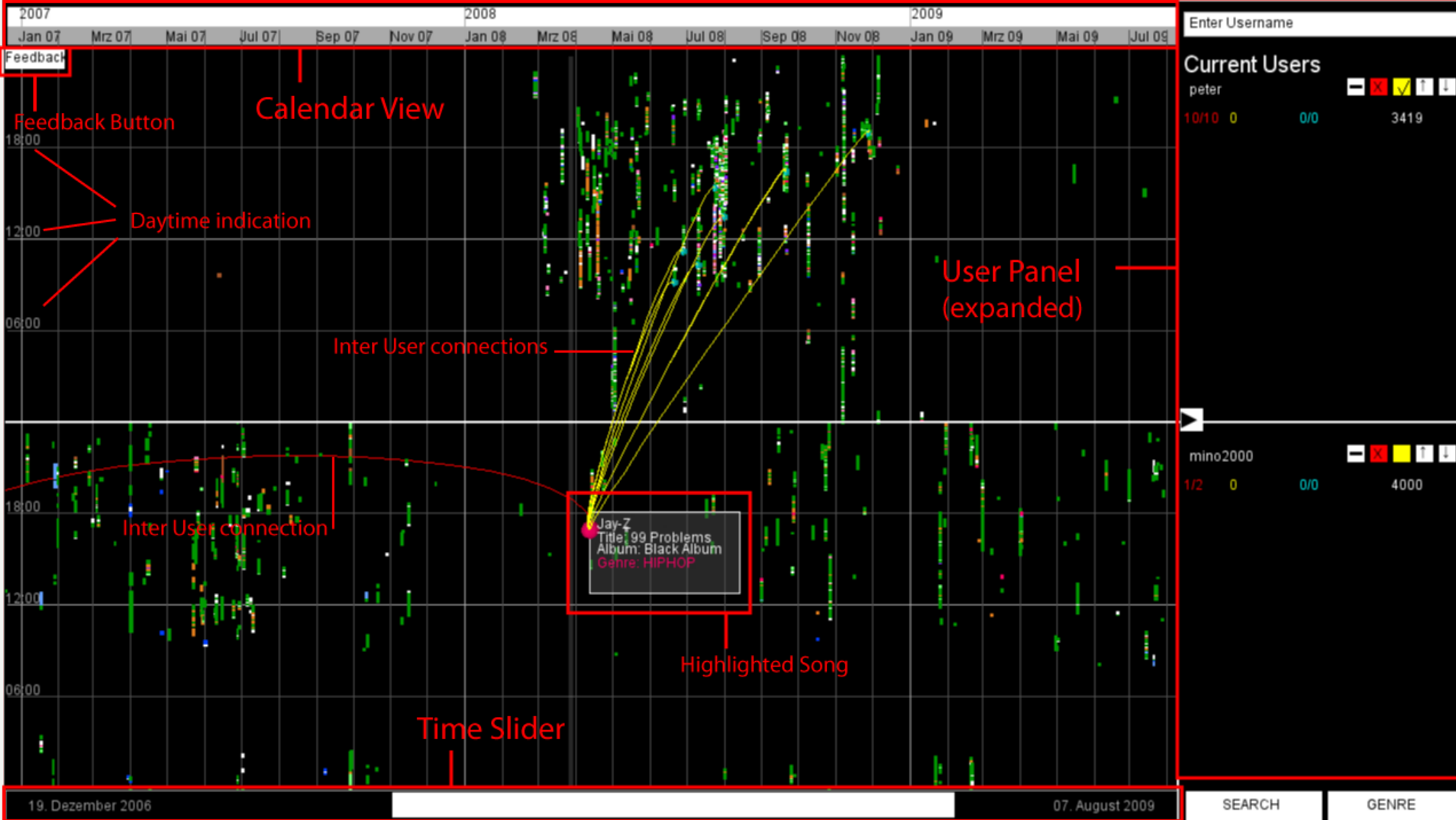


LastLoop




Ziele und Anforderungen

- direkten Vergleich zwischen multiplen Benutzern ermöglichen
- Temporäre Abhängigkeiten hervorheben ⁽¹⁰⁾
- Interaktiv ^(8,9,10)
 - u.a.: Navigation, Suche, Filter...
- Details bis auf Liedebene ⁽¹⁰⁾
- ausreichende Performanz



Implementierung

- Java Applet
- API's: Processing ^(5) , last.fm bindings for Java ^(6) , JavaMail ^(7)
- Speichern von Datensätzen auf HDD
- Multithreading: Laden der Last.fm Daten im Hintergrund
- Programm ist Open-Source 





Evaluation

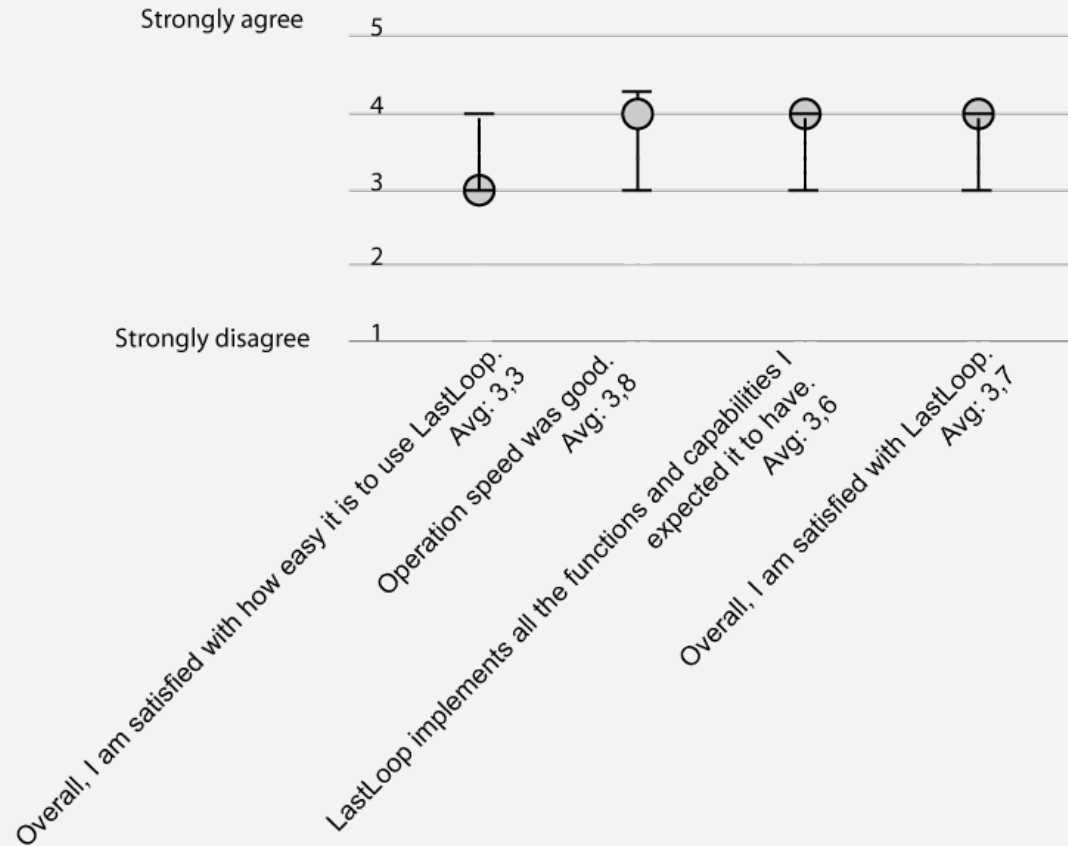


Online Survey^(8)

- Auszufüllen nach anschauen von Video + Benutzen der Anwendung
- Quantitative & qualitative Fragen
- 20 vollständige Antworten

Quantitativ

- Anwendung gut angenommen (i. D. ~3.4 / 5)



Qualitativ

„The way I listened to music was deeply related to the things that were happening in my relationships.“

„That one user is also listening to a very unfamous band, from the 70th“

„Interuser connections showed interesting facts: when did the other user hear my favourite song, have there been many connections lately, ... ?“



Ausblick



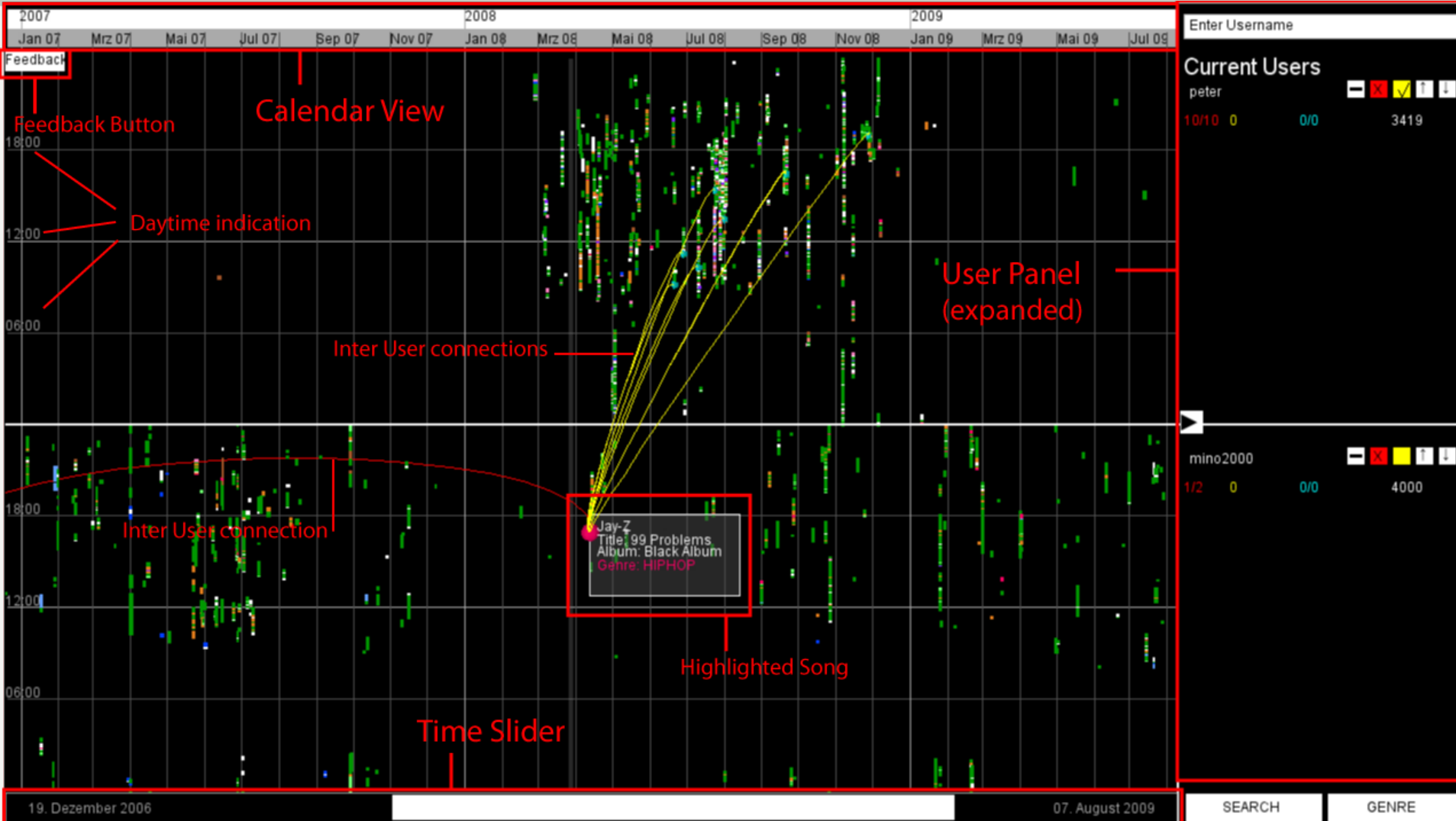
Mögliche Verbesserungen

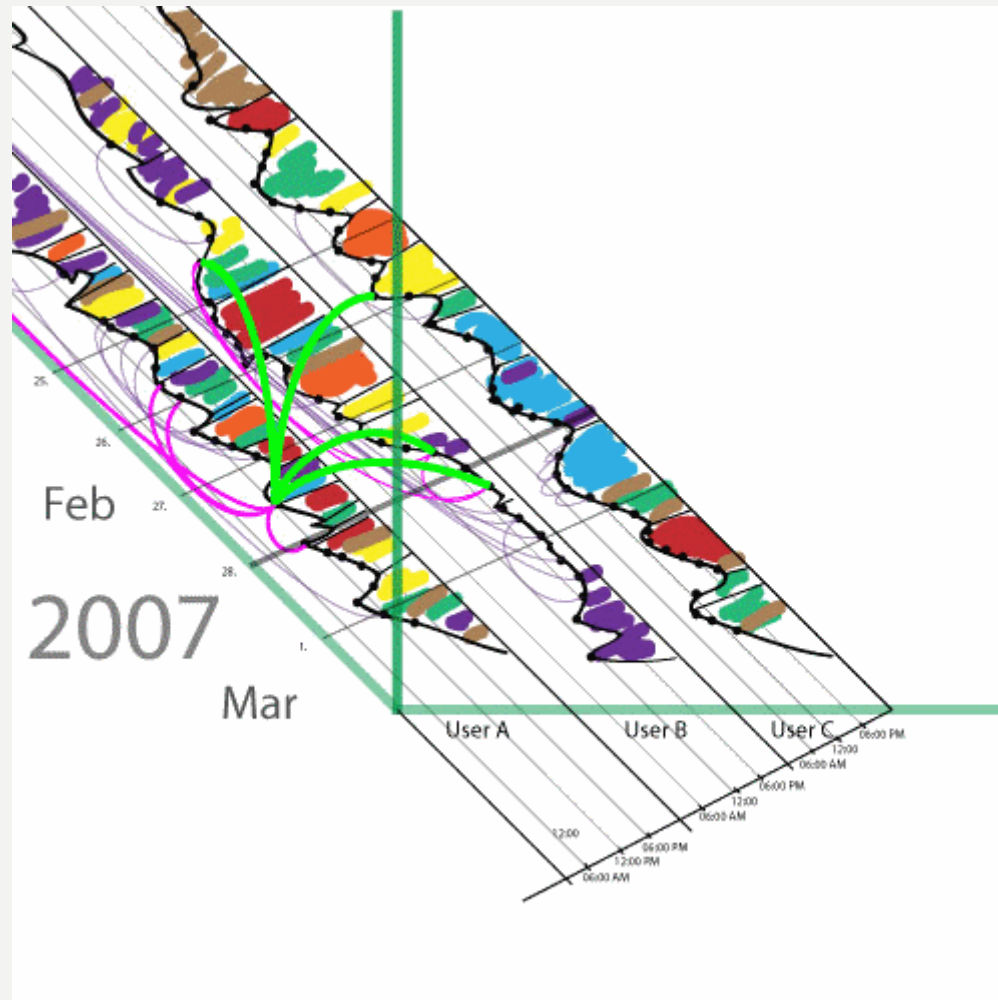
- UI verbessern
 - Liedauswahl wenn herausgezoomt (Lupe?)
 - TimeSlider überarbeiten
 - mehr “Schließen” und “Enter” Buttons inkl. Tooltips

- Visualisierung
 - Semantic Zoom verbessern (Performance)
 - (Wählbares) Limit für Songs
 - Dynamische Auflösung

Vielen Dank für ihre Aufmerksamkeit!

www.lastloop.de





TrackLibrary {final, serializable}

```

- vectArtist: Vector<String>
- vectTitle: Vector<String>
- vectAlbum: Vector<String>
- vectGenres: Vector<Genre>
- lookUpArtist Map<String, CopyOnWriteArrayList<Integer>>
- lookUpTitle Map<String, CopyOnWriteArrayList<Integer>>
- lookUpAlbum Map<String, CopyOnWriteArrayList<Integer>>
+ getArtist(songId: int): String
+ getTitle(songId: int): String
+ getAlbum(songId: int): String
+ getGenre(songId: int): Genre
+ searchTitle(text : String):CopyOnWriteArrayList<Integer>
+ searchAlbum(text : String):CopyOnWriteArrayList<Integer>
+ searchArtist(text : String):CopyOnWriteArrayList<Integer>
+ getGenres():Vector<Genre>
+ setGenre(int: indexAt, genre Genre):void
+ addSong(track: Track):void
+ alreadyExists(track: Track):int
+ saveTrackLibrary():void
+ loadTrackLibrary():void

```

User {serializable, observer}

```

- username: String
- userId: int
- timeToSong: ConcurrentNavigableMap<Long, Integer>
- songToTimes: Map<Integer, CopyOnWriteArrayList<Long>>
- intraUser: ConcurrentNavigableMap<Long, Long>
- interUser: ConcurrentNavigableMap<Long, CopyOnWriteArrayList<User>>
- lfmCon : LastFMCon
- userVis: UserVisualization
...
+ addTracks(tracks: Collection<Track>): void
- addInterUser(track: Track, userB: User, int: songId): void
~ cleanInterUser(remUser: User, songId: int)
+ getUserName(): String
...

```




Pulling Strings from a Tangle⁽²⁾

Tangle: Node-Link Diagramm



Strings: Darstellung von Sessions



- 1 - D. Baur: **Visualizing Media and Music Histories. 2010**
- 2 - D. Baur, A. Butz: **Pulling strings from a tangle: visualizing a personal music listening history.(2009)**
- 3 - F. Seiffert: **LastHistory - Visualizing Last.fm Listening Histories and Personal Streams**
(<http://www.frederikseiffert.de/lasthistory/>), accessed August 29,2010
- 4 - A. Turnlav: **Last.fm Explorer** (http://alex.turnlav.net/last_fm_explorer/), accessed September 12, 2010
- 5 – B. Fry, C. Reas: **Processing** (<http://processing.org/>), accessed September 12, 2010
- 6 - J. Kovacs: **last.fm API bindings for Java** (<http://www.u-mass.de/lastfm>), accessed August 29,2010
- 7- **JavaMail API** (<http://www.oracle.com/technetwork/java/index-jsp-139225.html>), accessed September 12, 2010

8 - Manuel Lima. Information Visualization Manifesto. <http://www.visualcomplexity.com/vc/blog/?p=644> (2009)

9 - Card et al. Readings in information visualization: using vision to think. (1999)

10 – Bade et al.: Connecting Time-Oriented Data and Information to a Coherent Interactive Visualization. In: *Proceedings of CHI'04, ACM Conference on Human Factors in Computing Systems*, ACM Press, pp. 105-112, 2004.