Nick Senske
Ryan Rindler
Michael Toulouse
Amy Grude
Voratima Orawannukul
Michael J. Wise
IOE536 Semester Project Final Report – Group 4
Instructor: Dr. Nadine Sarter
Submitted 12/13/2006

Abstract

Apple Computer's popular iTunes software has been frequently lauded for its elegant design and ease of use in managing all forms of digital media. However, despite adherence to Apple's "Human Interface Guidelines", the iTunes interface still can create many problems, especially for inexperienced users. The design team focused on two specific areas for improvement, addressing shortcomings in how iTunes searches for and organizes music (or other media), as well as the overt attention requirements for playback control. For the first problem, a search visualization tool called iBrowse that would display music in a more meaningful and manageable network format is proposed, facilitating both searching and browsing for the user. Additionally, the design group developed a 'smart' playback functionality called SmartPlay that monitors the user and automatically adjust the music to support his changing moods and activities. A second 'party mode' aspect to SmartPlay allows a user to invoke and aim for the creation of certain moods.

iTunes Introduction

As music lovers around the world enter the digital age, more and more support has grown around Apple Computer's iTunes application. Since it was first introduced in January of 2001, the iTunes software has been downloaded over 100 million times, making it the most popular digital jukebox around (Cook, 2005). But iTunes doesn't just play music; it also provides the ability to search, sort, and organize music, share songs with friends, burn and rip CDs, interface with the portable iPod music player, listen to the radio, and even purchase new music. Recently, this play, sort, share, shop functionality has expanded to other forms of media including podcasts (recorded broadcasts), music videos, television shows, movies, and games. iTunes has established itself as a central hub for all digital media.

The iTunes interface has also been embraced and praised for its simple and elegant design. Certainly converting huge stacks of CDs and LPs into clean, alphabetized lists is a huge step forward, but iTunes has taken it even further. Apple's "Human Interface Guidelines", much like Shneiderman's interface design principles (Shneiderman, 1998), include direct manipulation, user feedback, consistency and forgiveness. Adherence to these guidelines helps to make iTunes one of the easiest applications to learn and use. However, this design is still tailored toward the seasoned computer user, overlooking much of iTunes potential audience.

The iTunes interface design incorporates many standard software features. Pull down menus, ordered columns, and search boxes are a few of the examples that facilitate music organization on this interface. But these designs are only standard to those with years of computing experience. For computer novices, many of the convenient organizational features iTunes offers appear confusing with no explanation. Lists and matrices are only valuable if the user can understand and manipulate them as they desire. This confusion only grows as errors

occur in the interface. Mislabeled, misfiled, and even deleted music can cause problems in organizing music. Even worse, a missing file can go undetected for days or weeks when staring at a list of thousands of songs. And in general, managing thousands of songs (as many users do) can be very difficult in a list format. Data overload certainly becomes a problem when scrolling through pages and pages of files, looking for one small detail.

The second major problem experienced with the iTunes interface is the active attention and overt action it requires to operate. While many users consider music as an environment-enhancing backdrop to their daily lives, all of the complex functionality has turned music appreciation into a very engrossing, time-consuming activity. The functional purpose of music is not buying or organizing it, but enjoying it, and probably enjoying it while doing something else. However, monitoring and changing music in iTunes often disrupts that primary task. Where iTunes is supposed to support the user's daily life, it is often the other way around. The interface design focuses on using iTunes, engrossing users in time-consuming activity, when it should really focus on helping to achieve user goals that live outside of iTunes.

Both of these problems will be explored below in greater detail, with thoughts on how these design obstacles could be remedied.

iBrowse: Problem Identification

An iTunes user that wants to find a particular song or artist in their music collection or in the iTunes store faces a potentially difficult task. A general search from the search box returns all songs that contain any metadata matching that keyword, which drastically increases the amount of unwanted returns the user must wade through to find what they want. This metadata can be related to genre, title, artist, album name, group, composer, and comments. In an

exploratory scenario, for example one in which the user wants to find out if iTunes has songs by a particular band, the system may return hundreds of entries not at all related to the original query, as is shown in Figure 1 below. In some cases, the user is not clear why the song matched the search criteria to begin with.

Figure 1: A User Search for the Band "The Recipe" Returns Over 150 Results. The First Page Does Not Yield Any Songs by the Band "The Recipe"

Heuristic evaluations of the search feature found that searches using metadata only were problematic for several reasons beyond the return of too many songs. First, the results shown are only those that match the metadata exactly, whereas the goals of the users (in this case, the three personas of Daveryn, Jonathan, and Tammy located in Appendix D) were often not that explicit. Often, the users were not exactly certain of the name of the band or song they were looking for, and typed in words that they thought might return results. Though iTunes has an advanced

search feature, novice iTunes and novice computer users such as the personas were not aware of it at all and did not consider the possibility that one might exist.

Another problematic feature found through heuristic evaluation was the function of the search box in relation to playlists. All users, including the expert iTunes user, experienced problems in which they performed a search which they assumed was global, only to realize that iTunes was only searching the playlist they were currently in, as shown in Figure 2 below. They switched to the correct playlist or to the full music library only to find that their search terms had been erased during the change. This was frustrating and exemplified the lack of location awareness held by the users.

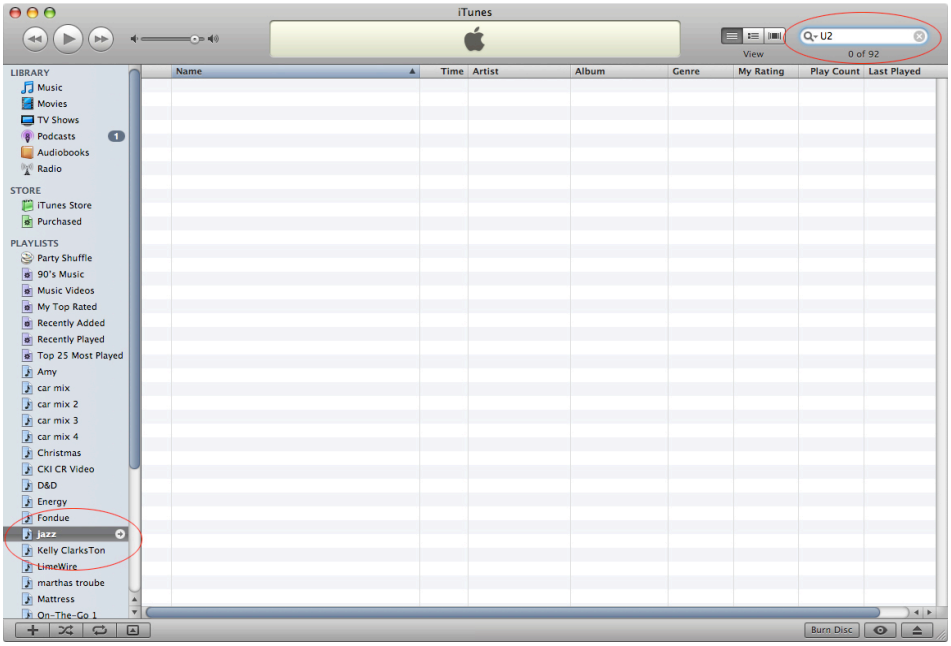


Figure 2. Unintentional Search Within a Playlist Yields No Results

Users attempting to browse iTunes (that is, search from a selection list of all possible genres, artists and albums), not knowing what they are looking for, have even more difficulty than those searching for something specific. First, it is not obvious that a browser exists or how to view it. This is something that realistically would only be discovered while looking through

drop down menus. Once the user does locate the browser (under the "View" menu), however, they are constrained by the provided filters of genre, artist, and album, in that order (as shown in Figure 3). Within the system, there are no discernable relationships between songs or artists that could lead to beneficial exploration. For example, if a user is planning an event such as a wedding, and wants to search their library for possible wedding songs, the only way to do this is to go through each song, one by one, and evaluate them on an individual basis.

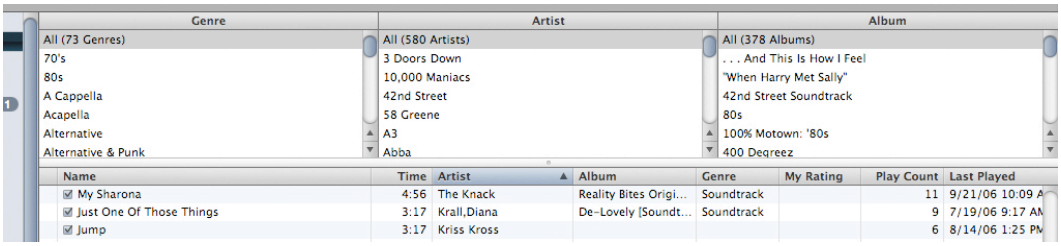


Figure 3: The iTunes Browser

This is only one example of many types of user queries that do not fit the current options available with metadata searching and browsing. Many songs lack full metadata, and therefore are not included in searching and browsing. Additionally, users have difficulty in searching and browsing when multiple artists are listed for a song. The "artist" category is not a tag-like infrastructure where multiple artists can all be "attached" to a song. Instead, the "artist" category is simply a label, and one artist can be included in multiple labels, as in Figure 4, below. Even songs or media that have only one artist may have that artist listed under multiple spellings or with different punctuation that separate them into different iTunes categories.

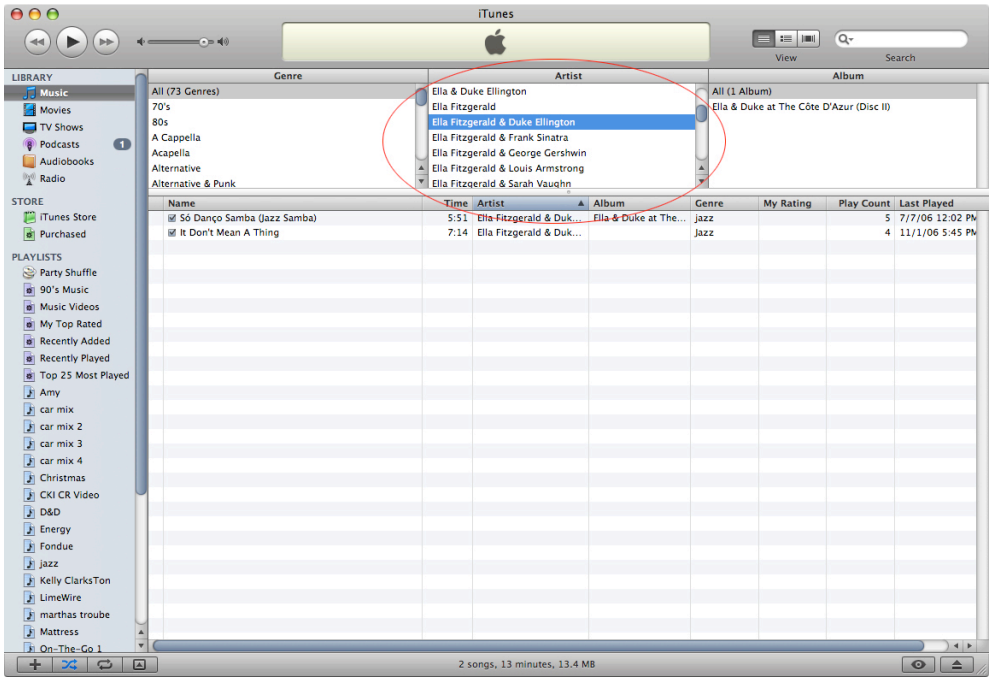


Figure 4: Inconsistent Artist Labels

iTunes does provide information about the current top downloads within each genre, but beyond that and a few highlights, there is no way to explore other than looking at lists. Within the system, there are no discernable relationships between songs or artists that could lead to beneficial exploration. When browsing, users follow nuanced paths between items; iTunes does not support this in any way.

iBrowse: System Description and Functionality

In designing an improved search and browsing system, named iBrowse, there are two basic types of recommender systems to be considered: content-based recommending and collaborative filtering. Content-based recommending is commonly known as a search engine. Current iTunes search and browse is considered a content-based since songs are categorized by artist and can be searched textually by song name or band name. However, the previous section showed that this sort of search system, especially if implemented in a basic manner, does not

support the user when searching a highly interrelated database. The current iTunes system fails to take contextual relationships between songs into account in its listings and thus little opportunity is given for users to explore their music collection in a manner that exploits song, artist, and album relationships.

In contrast, collaborative filtering is a recommender system that gives personalized prediction based on similarity or correlation between users or products. For instance, people who like similar movies in the past tend to like the same movies in the future. Or, for another example, a customer who buys milk, bowl, and a spoon has high tendency to buy cereal as well. Within collaborative filtering, there are various approaches, which is domain dependent. This means some algorithm can predict a certain kind of item better that the other. The commonly used algorithms are item-to-item and user-to-user. Amazon, for instance, uses item-to-item corroboration to recommends "Customers who bought x also bought y." (Linden et al., 2003)

An additional problem with content-based search is that users cannot expand their music collection without having to know specific information i.e. artist, album, song's name, and genre. This is of special interest for Apple to improve since the iTunes store is the only revenue-generating portion of iTunes. If it is easier for users to find new albums and songs that are backed by a solid recommendation system, both Apple and users win. Plus, Boolean search (content-based) results in binary hits and misses; users must come up with not only the correct spelling but also the correct terminology. Also, iTunes' list view, artwork view, and album cover view allow only linear browsing; it is difficult to picture the relationship between songs. This is essential when users want to setup a playlist for a certain mood such as party or chill out. To enhance what iTunes currently provides, a combination of collaborative filtering and information visualization should be integrated to the system.

In the case of expanding a music collection, iTunes five-star rating, the current rating mechanism, is the best available mechanism for integrating a recommender system. Only the database is needed on the online music store to collect users' ratings, which results user by item metric. Once a user requests a recommendation for a song, the system will search for songs within the metric that are highly correlated with the input song. The recommendation calculation that was developed is shown in Appendix E. Users retain the option to limit the recommendation feature to certain genres, artists, or song speeds, for example, by selecting the appropriate branch off of the original song or album (see Figure 5), or they may simply allow the recommendation calculation to find the most closely related and highly rated songs compared to their starting point by choosing to expand the view without adding any additional constraints.

There are generally two forms of recommendation: (1) a rating for a particular song or (2) a list of highly recommended songs. The former can be used as an indicator how much users will like a song. The latter is presented as the 4th view on iTunes in form of iBrowse (see Figure 5). It is presented as an inline alternative to the basic list, artwork, and cover art views that are provided with iTunes.

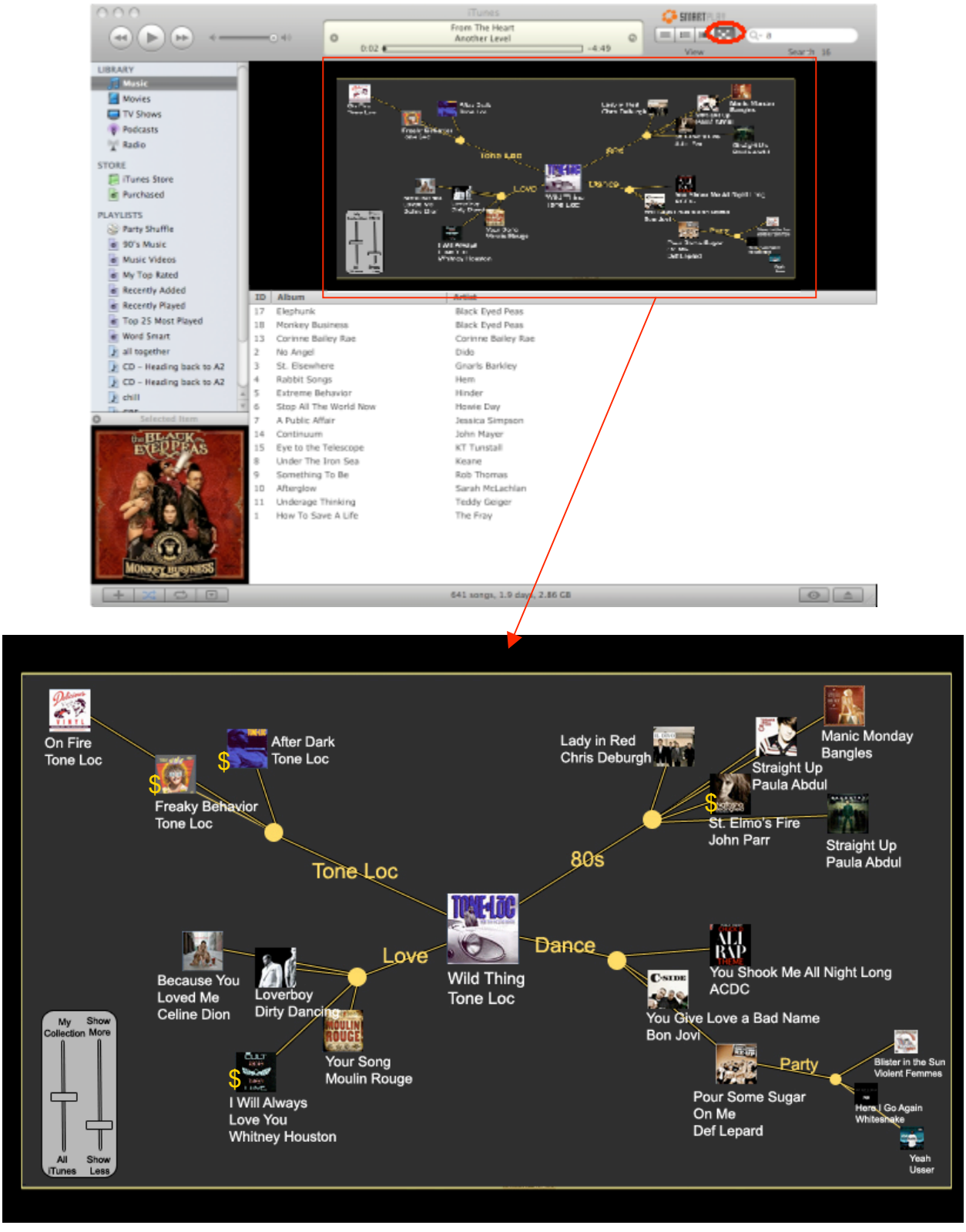


Figure 5: iBrowse Design Prototype

Each node represents either an album or a song. A link shows a relationship between nodes. The farther apart (less correlated) nodes are shown next to each other, the lower the relationship is between two nodes. No link means there is no relationship. On every node, users can expand to see more related albums or songs by placing the cursor over its picture, which gives to expand that view and explore related items or to close the album entirely and eliminate it from view. Multiple items may be dropped and dragged simultaneously onto this view and recommendation linkage will be shown among those items, if any exists.

Songs or albums can be dragged into the graphic network view from the traditional search list results shown below the graphic view to begin traversing their collection and possible purchases that can also be made through targeted linkage to the iTunes Store. Items available for purchase but not currently owned contain a yellow "$" (or other regionally appropriate currency symbol). Users will be prompted with a dialog before being asked to purchase anything. Also a slider is provided so as to allow the user to adjust to what extent new items from the iTunes store will be presented versus searching only one's own library.

This iBrowse network view dynamically resizes to accommodate the breadth of search the user is undergoing. It is also possible to simply pan around the display by clicking and dragging the mouse, similar to many other programs. Selections can also be dragged into playlists or double-clicked to be played immediately, consistent with the operation of the rest of the iTunes interface. The second slider presents a way to limit the number of potential linkages that are shown off of each object.

SmartPlay: Problem Identification

The second major problem noted by the group was a fundamental shortcoming with the design of iTunes. All of the walkthroughs and heuristic analyses emphasize that all interaction with iTunes necessarily must occur through its graphical user interface (GUI) on the host computer. Furthermore, the iTunes playlist is entirely dependent on direct user choice of the song(s) the user wishes to listen to. This is troubling since consideration of the users' goals (and consideration of all three personas Daveryn, Jonathan, and Tammy, located in Appendix D) indicate that their usage of iTunes is not one consisting of full-time user fixation on the graphical iTunes interface, but rather, for their primary goals, iTunes is generally expected to perform in the background.

In other words, the team's personas wish to accomplish tasks on other parts of the graphical user interface of their computer, not simply focus on giving iTunes specific direction on what songs to play and when to play them. Modern GUIs available on desktop computers have few avenues for applications that largely function in the background but may require continuing user input or inquiry as a secondary task without incurring relatively major time and attentional costs. Graphical interfaces also tend to be highly deterministic in their input requirements for users and do not allow for probabilistic input that may be unclear in its meaning. For example, target areas to initiate actions on a GUI are defined almost exclusively as hit or miss areas. There is no probabilistic consideration of input that may not be purely deterministic in nature. (Voice dictation, still in its relative infancy, is a good example of how difficult probabilistic interpretation of user input is.)

Figure 6: Apple Remote Control

iTunes currently offers playback control via the iTunes GUI and with a simple infrared remote (shown in Figure 6) only available with certain new iMac computers and notebooks. Task switching on a GUI is time-consuming and likely to frustrate a user that is actively engaged in other work. The remote is in the same vein of other audio/visual controls that have shown little innovation for decades and only offers simple, direct playback controls. While remote controls are convenient and remove some task-switching issues with the GUI, they offer little feedback and still rely entirely on the deterministic input of the user. The sparseness of the option presentation and feedback it provides little or no decision support for the user and it essentially only moves a subset of functionality off of the GUI but does not enhance it in any meaningful way.

The extensive playlist management support of iTunes only works through users' explicit selections. The only non-user controlled playback choice mechanism is offered through a random song selection, which is in effect an anti-intelligent method for choosing songs since it explicitly takes nothing about user preferences into account when choosing songs to be played except restricting what it plays to the playlist the user has chosen. The "Smart Shuffle" feature (shown in Figure 7) allows users to make it more or less likely certain albums or songs by certain artists can be played, but this functionality is buried within the preferences menu and is still

essentially random with a few added minor constraints to the randomizer's behavior. Finally, users can give ratings to song and organize songs via those ratings, but again this relies on users' explicit, overt actions.

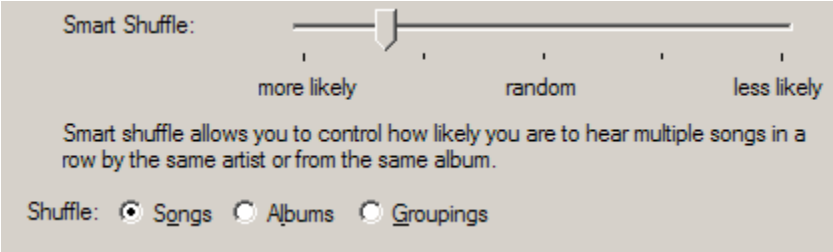


Figure 7: iTunes Smart Shuffle Feature

The above detailed issues with this fundamental limitation of iTunes led the team to realize the need to ameliorate the sparse and deterministic nature of the input that iTunes was receiving regarding the user's preferences and desires, especially since music is highly tied to mood and emotion. An affective understanding of the user must be supported by iTunes to better support the user experience.

SmartPlay: System Description and Functionality

'Mood Supporting Mode'

As stated earlier, for the majority of users and for all of the personas generated for this project, iTunes functions in the background while other tasks are performed. Even a simple action like changing a track can shift attention away from these tasks and decrease productivity. To address this problem, SmartPlay is a new function, integrated into iTunes, which intelligently manages a user's playlist with the intention of preserving a state of 'flow' in their work. The goal is to provide the best music possible with a minimum of intervention (see Figure 8, below).

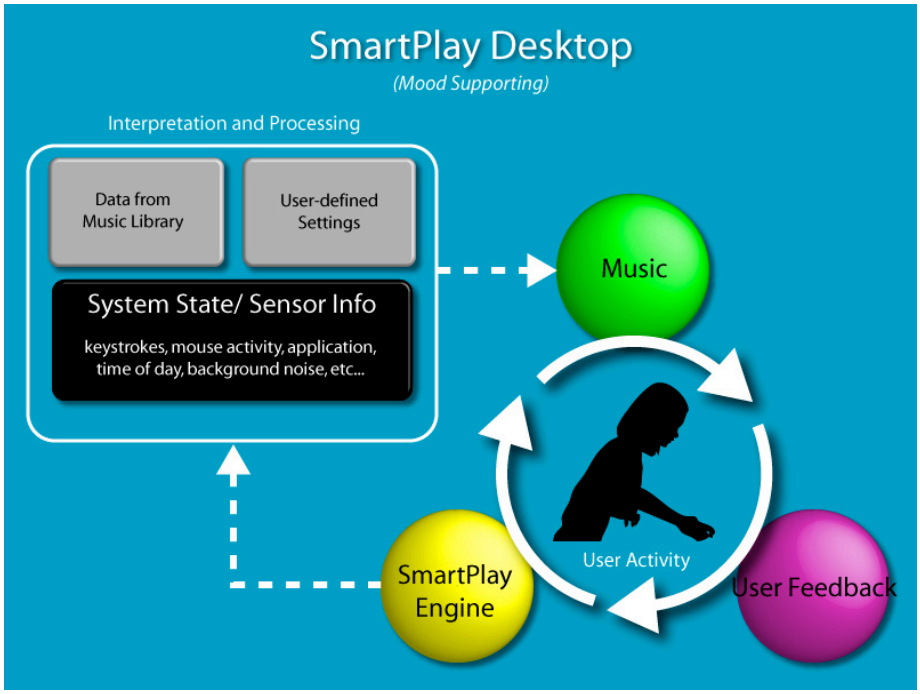


Figure 8: Conceptual Diagram, 'Mood Supporting' SmartPlay

SmartPlay accomplishes this with two interacting systems. First, SmartPlay has an artificial intelligence engine that monitors and responds to several measures of user input. For example: the time of day, applications open, speed of program switching, input speed and variability, etc. When SmartPlay is first activated, users continue to use iTunes as normal. During this calibration period, SmartPlay develops a baseline profile of the user's music habits while it records the state of their system and their interactions. The system also processes the user's music library for track data such as beats per minute, volume levels, average pitch, etc. The calibration data and track information are combined into a database that the engine uses to determine what music to play and when.

When active, the system tracks activities such as keystrokes per minute and scores them. These scores are checked against the database where they are associated with song characteristics such as beats per minute. In this manner, the system reads the user's interactions and dynamically produces playlists as needed. If the user's interaction profile remains the same,

the characteristics of the next song will be similar to the current one. If there are changes in the data such that it no longer fits the current profile, SmartPlay switches to another group of songs that better matches the usage pattern. Other computer usage metrics that could potentially be integrated into this functionality that helps classify and identify user preferences include mouse speed, mouse clicks per unit time, or active computer process (e.g. web browser, Microsoft Word, etc.)

This system alone might be somewhat successful, but invariably errors will arise with stereotypical interpretations of implicit user input and thus demand the user's attention, defeating the purpose of this system. It is impossible to determine what music a person prefers from environmental data alone and such failures will greatly increase the chance such a feature will go unused. Therefore, the second component of SmartPlay addresses these potential shortcomings. Selecting its icon with no further input from within iTunes activates this mode of SmartPlay. The user then has the option to give it direct feedback. The function is designed to be simple and involve minimal interaction from the user[1]: (Apple Symbol or Alt) + ↑ for 'I like it' and (Apple Symbol or Alt) + ↓ (Command + Down arrow) for 'I don't like it'[2].

The short-term effect of this operation is to change the kind of music being played. SmartPlay will automatically attempt to find the next most appropriate music based on the database. If it is not successful after many attempts, it may reach the point where it suggests

[1] Simple, because it is not too specific as to require much thought. It is a binary choice. Ideally, the user can perform the operation without interrupting their work. The keyboard keys were chosen because they map well to the feedback (up for 'yes', down for 'no') and they do not require switching to mouse input while the user types. Alternately, for programs that use a combination of the mouse and keyboard (such as Photoshop), Apple-Symbol or Alt + mouse scrolling could substitute.

[2] New users of SmartPlay are greeted with a dialogue describing this process the first time they attempt to change a track manually. This is the appropriate time to teach them about the feedback function. Later, these instructions are occasionally displayed scrolling across the SmartPlay dock icon as a reminder. Rating a song in this manner serves as a shortcut to enable this mood-supporting mode of SmartPlay if the user has not already enabled it directly on the iTunes interface. Manual selection of songs from within iTunes indicates to the system that SmartPlay is to remain inactive until the user invokes this rating system again or directly enables SmartPlay.

some completely different music or possibly some new music from the iTunes Store. At the same time, this feedback trains SmartPlay to make a better prediction the next time a similar situation is encountered. Over time, the user's profile becomes more and more specific and allows for more accurate predictions to be made as new relationships between interaction and music are recorded. Thus, through a combination of user tracking and explicit feedback, the system adapts to fit the user's music patterns.

'Party Mode'

One of the growing trends in personal computing is the transformation of the computer from a primarily work-centered machine, as described in the previous section, to that of a home entertainment center. For example, Windows has a Media Center Edition and Apple's latest computers feature a program called Front Row. Both editions contain special recording and playback software and include a remote control so users can operate their computer like a stereo or a television set. iTunes is also part of this home entertainment trend, as the program makes use of the Apple remote. For this reason, the design group decided to address the usability and attention requirements of iTunes in this context.

The previous example introduced the idea of SmartPlay on the desktop, serving a mood-supporting function, but the system could also be extended to home entertainment. Consider how iTunes might be used in an environment with many people at once. Imagine someone is hosting a party; one of their primary responsibilities is music. Typically, the host begins with a preloaded playlist for the guests to dance to, but every so often a song is not well received and the dancing trails off. Ideally, the host should be able to fix the song without shifting attention away from his or her guests. Even with a remote control, changing tracks is a distraction and offers limited control. If the mood of the party dramatically changes, even more attention may be required to

load a completely different playlist. In this kind of situation, a system like SmartPlay would be useful (see Figure 9, below).

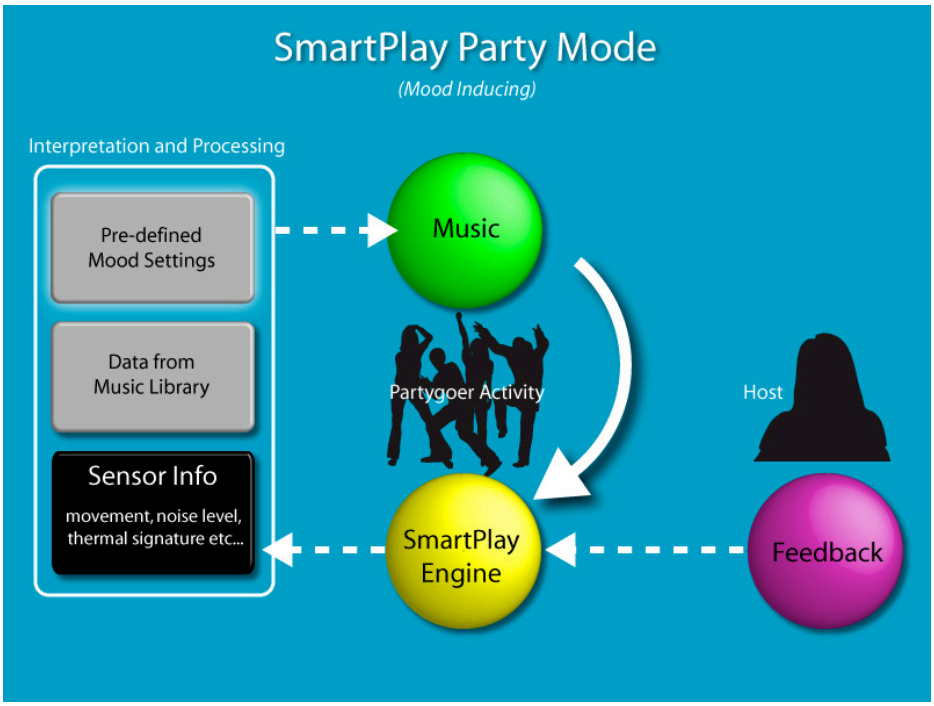


Figure 9: Conceptual Diagram, SmartPlay 'Party Mode'

With some minor adjustments, the desktop system could be adapted to home entertainment. However, since the focus of activity is no longer the computer, this scenario would involve the addition of sensors into the environment. Simple audio sensors could determine the frequency and volume of background noise, a thermal camera could pick up body heat, population density, and movement, and floor sensors could capture motion and dancing beats. With this information, SmartPlay could interpret the mood of the party and adapt the playlist to match. Finally, like the desktop system, some element of feedback is essential for improving the success rate and overall usability. For this function, the host could employ the iTunes remote or something similar. Alternately, some other interface types might be more useful, such as listening for audio cues (squeals or groans), cell phone applets, or specially designed pieces of furniture.

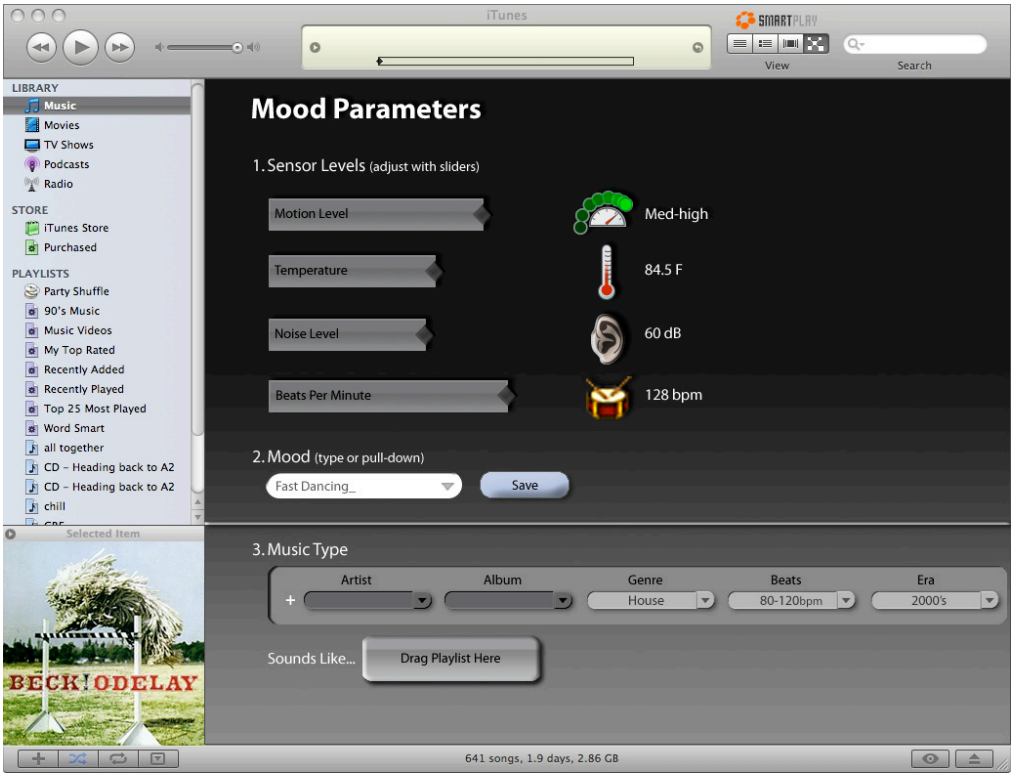


Figure 10: Mood Setting for Party Mode, within SmartPlay

In addition to its predictive capacity, the host could set SmartPlay to invoke a certain mood. This setting could come from a pre-generated list of moods (e.g. excited, conversational, intimate) with associated sensor thresholds, or through a special interface where mood criteria are explicitly specified (e.g. “the crowd should be louder than 100dB and moving on the dance floor at a minimum of 145 beats per minute”). In practice, the host opens up the mood dialog in iTunes, selects a setting, and starts SmartPlay, as shown in Figure 10 above. Then, rather than have the user sort through all their tracks manually, the system creates a playlist based on this mood. During the party, SmartPlay monitors sensors in the room and adjusts the music to gradually steer or otherwise maintain this mood (defined as a range of sensor data) among the guests, such as in Figure 11, below. This also shows how a visualization could be provided to gauge the mood of the party even if not in directly at the computer itself through abstraction of a camera image and associated color-coding. In this manner, SmartPlay could be extended into

home entertainment with the added functionality of *inducing* mood, as well as supporting mood. Once again, the system supports the needs of the user with minimal interaction and the capacity for adaptation and improvisation.

Figure 11: 'Party Mode' Monitor View

New System Evaluation Methods

The authors, as the designers of this new system, will have inherent familiarity with the system that they are designing. In order to accommodate users outside of the project team, it is important to elicit outside feedback all the way though the design process. This feedback should be gathered in an iterative process.

In the first stages of development, surveys would look for initial ideas on areas where the existing iTunes system could be improved. This would be preferred to the strategy used in this actual project where targets had to be selected without direct user guidance such as a survey. In

these surveys, bias would be avoided as much as possible by surveying a random but representative sample of the population of iTunes users. From the results compiled from the surveys both structured and unstructured interviews. These interviews would have to be limited at first to the functionality that was already developed (to avoid negative responses to the unfinished product) but would grow broader later as the interface develops.

In order to complete final testing for the proposed interface and system changes, a strategy similar to that employed by Microsoft's Windows Vista launch could be used. The beta version of the product would be made available to select groups of people with the condition that they agree to give the data that they generate through usage of the new system back to the development group to further product development. In this way, the development group could ensure mass exposure to the intended market to guarantee that the vast majority of the intended audience would be able to use a near-final rendition of the product.

Conclusion

These enhancements to the iTunes interface present many improvements to iTunes with minimal additional user investment. SmartPlay's party mode ensures that the user's guests will enjoy themselves while listening to movement inspiring music, while its mood supporting mode enables hassle free, automatic music selection, allowing the user to listen to great music without being interrupted from his or her work. The new iBrowse search and browse visualization tools allow the user to quickly find the song that he or she is looking for while the recommendation engine suggests songs that the user may wish to purchase to add to their collection.

These enhancements to the iTunes interface are more adaptive to the group's personas' goals and desires. It better understands their language in its allowance for greater spatial

expression and its derivation of mood and other parameters from implicit input. Also, SmartPlay's ability to adapt its own behavior depending on the user's approval of its actions allows for the development of an affective agent that knows best how to intervene for the user without requiring explicit direction. This logic results in advantages for both the user, in terms of greater listening enjoyment, and the provider of the interface, by increasing revenues generated from song purchasing.

iBrowse's recommendation engine, as well as SmartPlay's features (such as party and ambient mode) represents a step towards the future trend that such systems are going to follow. As the workload placed on the average person continues to increase, systems that feature automation as a way of reducing this workload will become increasingly prevalent.

These enhancements, however, must be designed with a note of caution. The cognitive aspects of a system must be considered in order for that system to remain viable to the general public. Future systems should be easy to use, intuitive, and able to predict the user's needs without interfering with presently un-automated tasks.

References

Cook, B. "AmTech Upgrades Apple Stock to 'Buy,' Ups Target Price to $63." The Mac Observer. Available from http://www.macobserver.com/stockwatch/2005/10/17.1.shtml

Linden, G., Smith, B., and J. York, "Amazon.com Recommendations Item-to-Item Collaborative Filtering." 2003, IEEE, pp. 76-80.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J. (1994). "GroupLens: An Open Architecture for Collaborative Filtering of Netnews." In CSCW '94: Conference on Computer Supported Coorperative Work (Chapel Hill, 1994), ACM, pp. 175-186.

Schneiderman, B. *Designing the User Interface: Strategies for Effective Human-Computer-Interaction*. Reading, Massachusetts: Addison Wesley Longman, 1998.

Smith, L. I., 'A tutorial on Principal Components Analysis', pp.1-8. Available from http://csnet.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf

Appendices

Appendix A – iTunes Uses and Information Needs (Submitted Electronically)

Appendix B – Storyboard Walkthroughs (Submitted Electronically)

Appendix C – Heuristic Evaluations (Submitted Electronically)

Appendix D – Personas (Submitted Electronically)

Appendix E – Recommendation Generation for Search/Browsing Visualization

Three steps are necessary to generate a recommendation. First, users are asked to rate songs, from which, the system can calculate relationships between each song (correlation coefficient). The value ranges from -1 (negatively correlated) to 1 (highly correlated). 0 correlation mean there is no relationship between the two items. The results from this step are used to create an item-to-item correlations metric. Finally, the system gives predictions to users. The more ratings in the metric, the better prediction, since more ratings means more knowledge about the relationships.

Table E1 is comprised of fictional data used to demonstrate this calculation. A blank cell represents an unrated song.

Table E1: users' ratings (user x item metric)

	$song_1$	$song_2$	$song_3$	$song_4$
$user_1$	2	2	1	5
$user_2$	3	4	4	2
$user_3$	1		2	4
$user_4$	5	3	4	
$user_5$	4	4	2	1

Step 1: Suppose User 4 is looking for a recommendation for Song 4. The system first generates a correlation coefficient using the following formula (Resnick et al., 2003) (Smith, 2002):

$$r_{i,j} = \frac{\sum_{i,j \in rated\_song} (song_i - \overline{song_i})(song_j - \overline{song_j})}{\sqrt{\sum_i (song_i - \overline{song_i})^2} \sqrt{\sum_j (song_j - \overline{song_j})^2}}$$

Step 2: The results are put together in Table E2, the item-to-item correlation matrix. Please note that the correlation coefficient between Song 1and Song 2 $(r_{1,2})$, and the correlation coefficient between Song 2 and Song 1 $(r_{2,1})$ yields the same value. For this reason, the matrix contains only half of the value; the second half is simply a reflection of the first half. Also, Table E2 omits the correlation between the song and itself, since they are highly correlated and always yield a value of 1.

Table E2: item-to-item correlation matrix

	$song_1$	$song_2$	$song_3$	$song_4$
$song_1$		0.4	0.6	-0.8
$song_2$			0.5	-1.0
$song_3$				-0.6
$song_4$				

Step 3: The prediction can now be calculated online using the formula below.

$$\Pr ediction(song_t, user_j) = \overline{song_t} + \frac{\sum_{i \in rated\_song\_by\_j} (song_i - \overline{song_i})(r_{t,i})}{\sum_{i \in rated\_song\_by\_j} |r_{t,i}|}$$

The prediction calculation of Song 4 for User 4 is as follow:

$$\begin{aligned}
\Pr ediction(song_4, user_4) &= \overline{song}_4 + \frac{\sum_{i \in rated_song_by_user4} (song_i - \overline{song}_i)(r_{4,i})}{\sum_{i \in rated_song_by_user4} |r_{4,i}|} \\
&= 3 + \frac{(5-3)(-0.8) + (3-3.3)(-1) + (4-2.6)(-0.6)}{|-0.8| + |-1| + |-0.6|} \\
&= 3 + \frac{-1.6 + 0.3 - 0.84}{|-0.8| + |-1| + |-0.6|} \\
&= 2.1
\end{aligned}$$

The prediction of Song 4 for User 4 is approximately two stars. An intuitive interpretation for the result can be made. Looking back at Table 2, Song 4 has negative relationships with the other tabs. Since User 4 seems to like Song 1, Song 2, and Song 3 (rated five, three and four stars consecutively), he/she might not like Song 4 as much. As a result, the recommendation is below Song 4's average score (3).

iTunes Intended & Unintended Uses, Information Needs

- Intended uses:
    - Listen to music
    - Watch videos, TV shows
    - Organize files into playlists
    - Organize/label music files
    - Search through music files
    - Download (Purchase) music/TV shows
    - Listen to music previews through iTunes store
    - Burning audio CDs
    - Burn backups of music files on disc
    - Share files with others on a local network
    - Convert music from CD to computer
    - Sync to iPods
        - Receive uploads of voice recordings from an iPod
    - Rating songs
    - File browser
    - Marketing research (player habits)
- Unintended uses
    - Browsing iTunes store with the intent of purchasing or downloading a file elsewhere
    - Using iTunes store as an informal database of music
    - Using a playlist for the purpose of file organization
    - Screensaver (visualizations)
    - Validation of one's preferences (e.g. looking at show/music popularity ratings so as to justify one's tastes)
    - Mass file renamer/organizer
- Information needs (*Italicized* needs appear to be unmet by iTunes)
    - File listings to choose from or search, method to choose files
    - Ways to invoke/control playback
    - Playback monitoring/feedback
    - Knowledge of what you want to play/watch
    - Location of files to add to library, how to add them
    - Knowledge of file organization on hard drive (if iTunes doesn't add all media files automatically)
    - Knowledge of where one is in the playlist/library hierarchy
    - How much hard drive space you are taking up and *how much space you have left*
        - How much space is used and available on iPod
    - File & album information/attributes
    - Playlist contents
    - What music is purchased vs. owned
    - Price, availability in iTunes store & how to purchase
    - Whose library/iPod one is accessing (multi-user systems)
    - System status (reading a disc, burning a disc, file transfers, etc.)
    - *Actual file/folder structure on hard drive*
    - *Recognition and grouping of similar artist names (e.g. The Smashing Pumpkins and Smashing Pumpkins)*

IOE536 Storyboard Analysis

Connecting and disconnecting an iPod to iTunes

By Nicholas Senske

The first time you use iTunes:

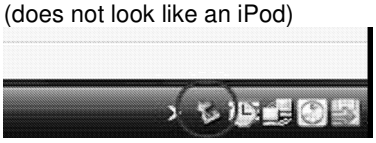
You must install the software before you use your iPod. Once iTunes is ready, you plug in your iPod and are prompted to synchronize it with your music library.

iTunes is necessary because music on your iPod is kept separate from your other data. Only iTunes can access and manipulate your music. (basically, it encrypts/hides it on your iPod) You cannot, for example, drag music from your file explorer directly onto your iPod and expect it to play. It must be processed through iTunes for this to work. Similarly, your other data cannot be read or organized by iTunes; you need two different applications. In effect the iPod functions as a hard drive or music player depending upon the application that accesses it.

1. Plug in your iPod; computer finds device; iTunes can be set to launch when connected

2. Default hardware icon appears in dock

(does not look like an iPod)



3. “Do not disconnect” appears on iPod

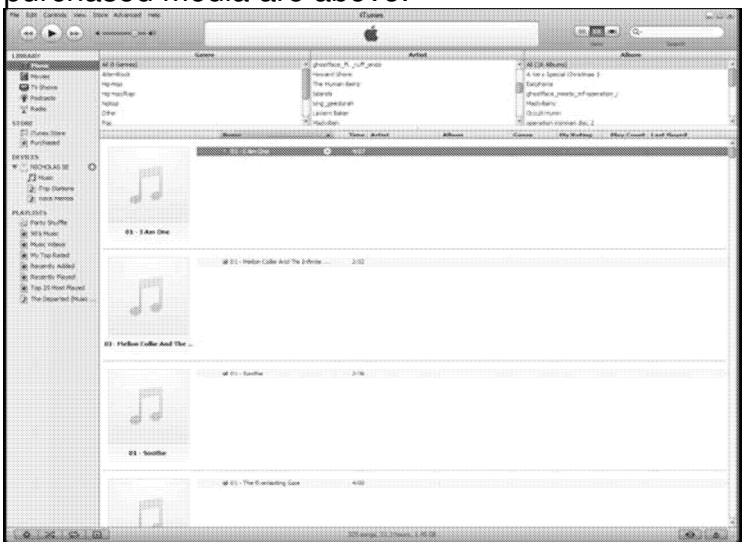
does not describe much to user.

4. Launch iTunes

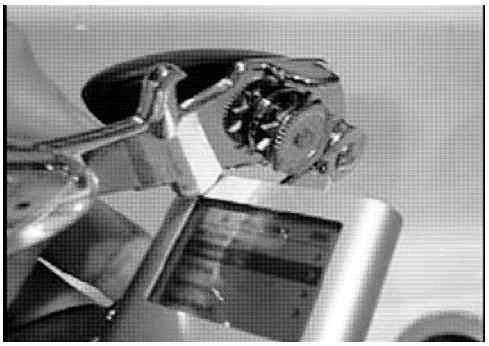
5. Summary screen comes up first;

shows device, options, space on HD, synchronization of music/ contacts/ podcasts/ etc.; some system information is duplicated at the bottom of the interface at all times

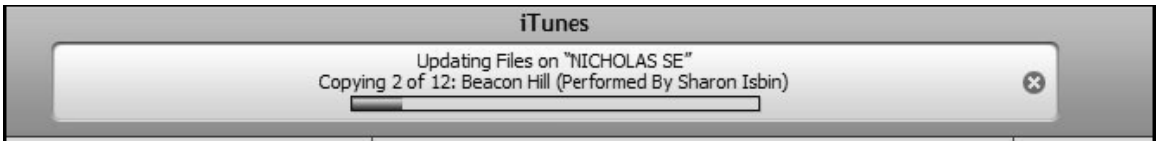
6. Throughout iTunes session, the iPod device remains in the dock on the left side of the screen. Your library and purchased media are above.



7. To transfer music or other media to the iPod, you first select the library, then the media you want (album, song, playlist, etc), and drag it over to the iPod device icon.
(you cannot transfer music off the iPod, only delete it)



8. A process window at the top of the screen shows the progress of the operation.

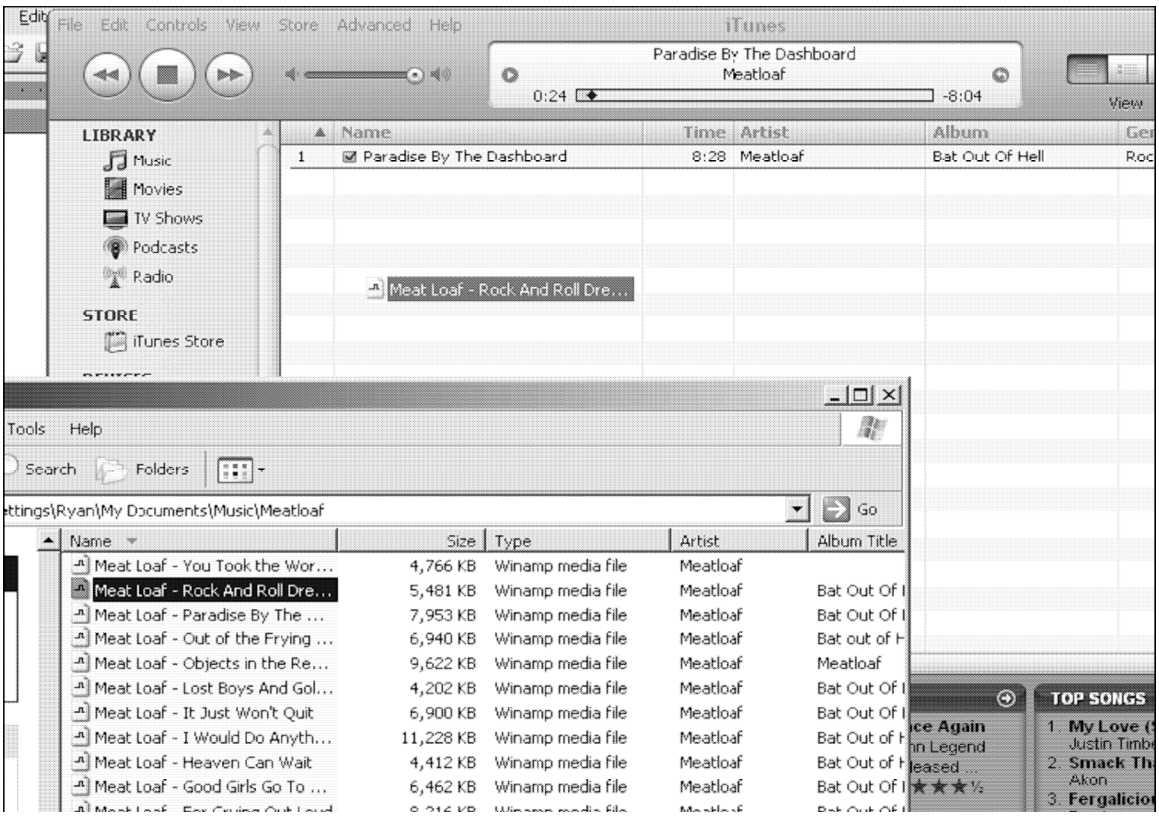


9. When you are finished, Quit iTunes
You must undock the iPod before you physically remove it.
You can eject your iPod using the iTunes or hardware icon in the dock, or with the icon within iTunes. (should it undock your iPod? or at least prompt you?)

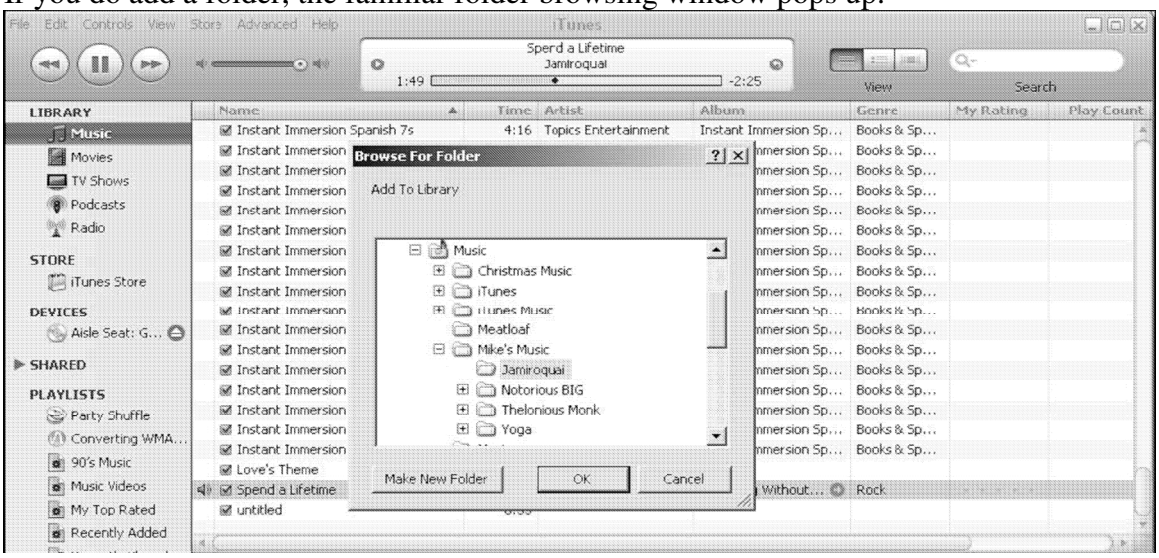
Inputting general digital music files, folders, and playlists to iTunes: (by Ryan Rindler)

From the File… menu, you can add either a file or folder of music to the library.

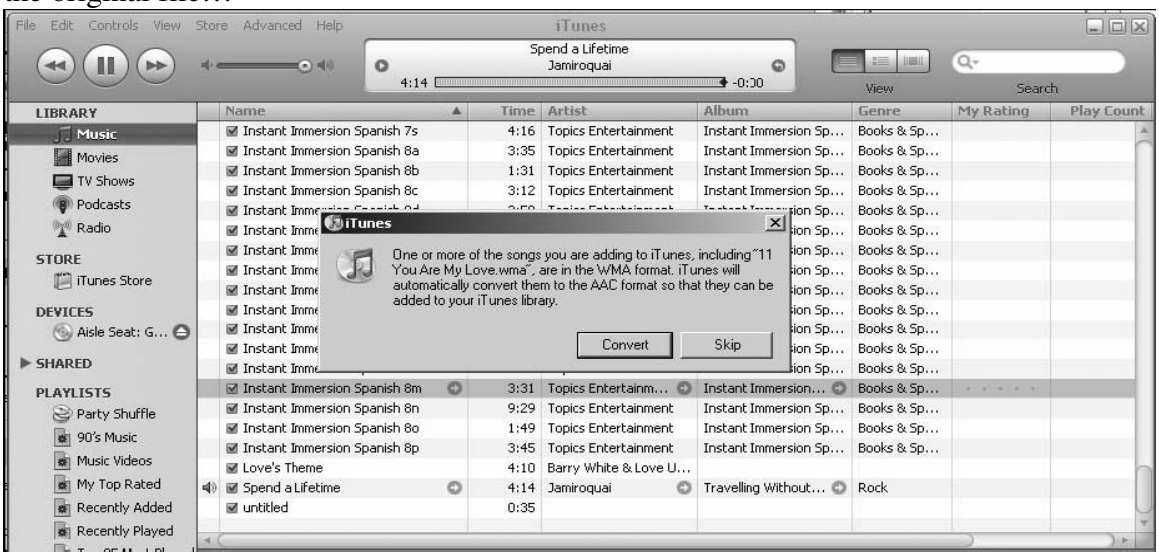
Or… similar to the standards of other GUI software, you can "drag and drop" music files



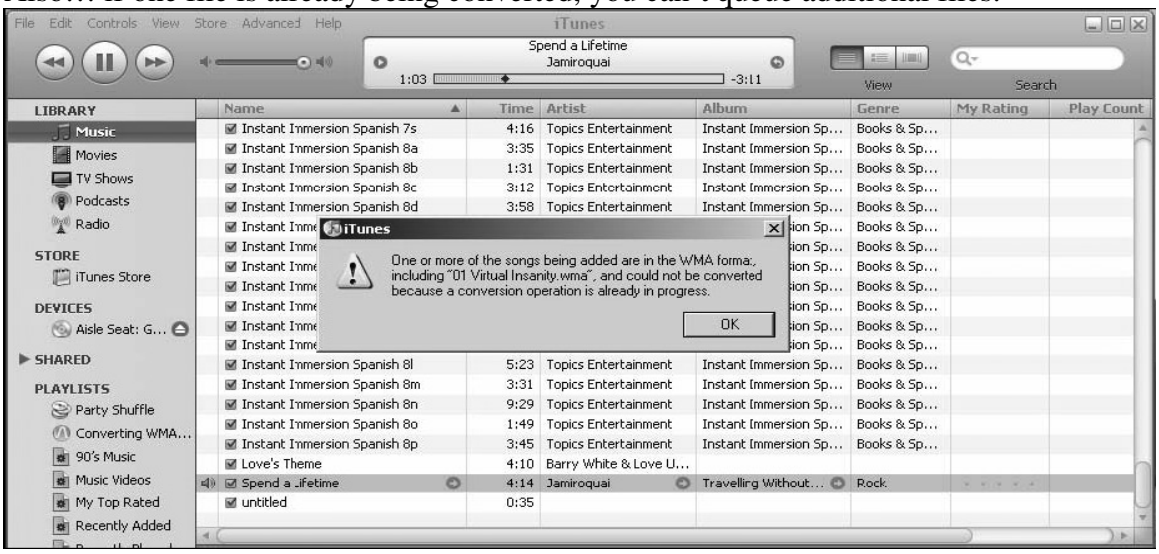
If you do add a folder, the familiar folder browsing window pops up.



However, if you add music of an incompatible format, a “warning” pops up, giving you the option (good!) to convert the file, but doesn’t specify if the format change will alter the original file…

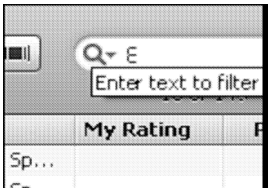


Also… if one file is already being converted, you can't queue additional files.



However, adding a folder of music to the library doesn't preserve the original playlist

You can use the search to view only certain files

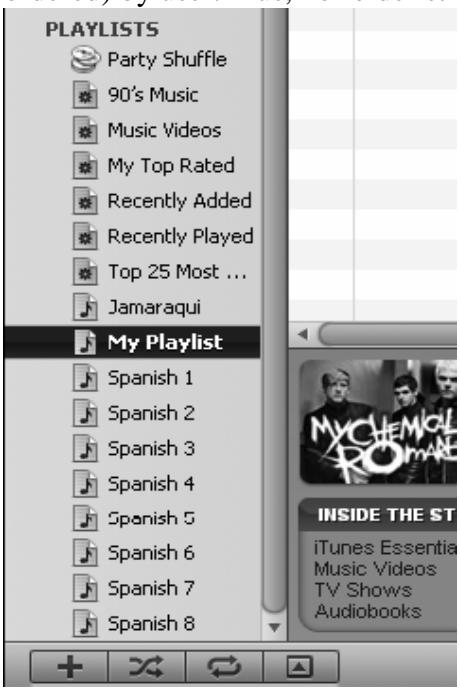


Otherwise… make a new playlist with File… New Playlist

You can then add the music desired to a given playlist (use right-click)

Or… “drag and drop” is once again supported.

Multiple playlists are organized alphabetically (good!), but cannot be changed (i.e. ordered) by user. Plus, no folder structure so 99 playlists will make a long left scroll-bar.

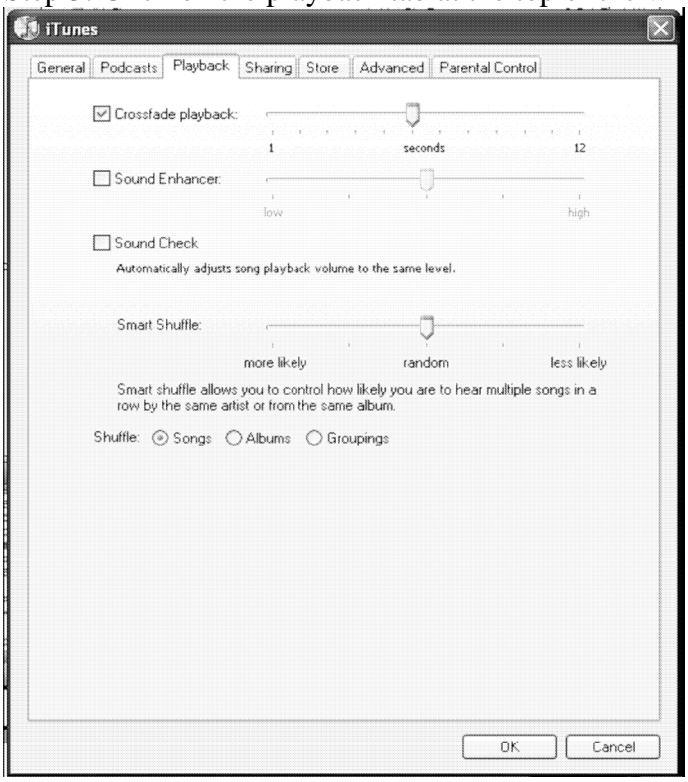


(By Michael Toulouse)
This story board covers the sound check feature of iTunes as well as the smart shuffle feature and the crossfade playback feature. The smart shuffle feature either increases or decreases the likelihood that songs from similar artists will be grouped together. The sound check feature normalizes your entire music library so all files retain a consistent volume. The crossfeed playback feature allows the end of the currently playing song to be mixed with the beginning of the next song in the play list so that no gap is present. (Once you implement these options, you'll need to shut down and restart iTunes for the setting to take effect.)

Step 1. Open iTunes

Step 2. Click on the edit menu and navigate to preferences

Step 3: Click on the playback tab at the top of the window that opens



Step 4. Enabling advanced features:

Clicking on the crossfade playback feature enables iTunes to mix songs so that when one song ends the second song will overlap it by a few seconds. (The slider adjusts the amount of this overlap.) Disabling this feature enables 100 % of a song to play and then starts the next song. A gap will exist between songs.
The Sound Check Feature when enabled modifies song playback volume to the same level for all tracks in iTunes.
The Smart Shuffle Feature controls how likely you are to hear multiple songs in a row by the same artist or from the same album. The radio button allows the choice between songs, albums, and groupings.

Notes:

It may be unclear whether crossfeed playback is enabled or not because some songs have recorded pauses or really quiet beginnings that are greater than the amount of time that the crossfeeder is mixing the two songs so that a gap may still exist.

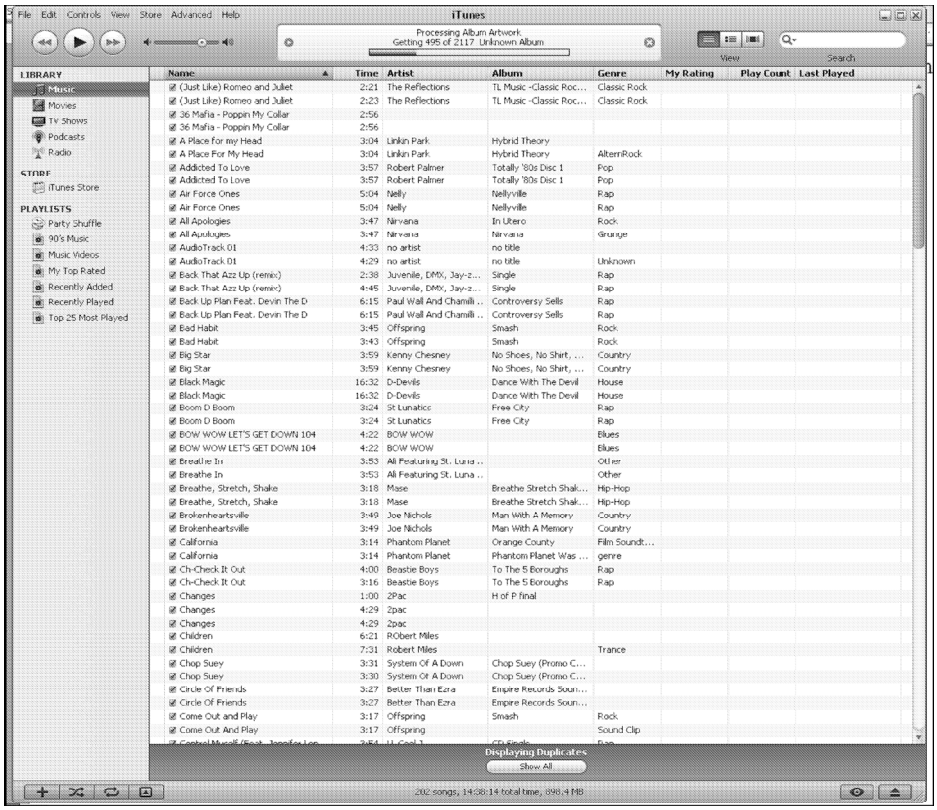
The Sound Check Feature does not give a user the option of picking the volume that the songs will be adjusted to. A possibility of biasing towards an abnormally high volume or an abnormally low volume may exist depending on the trends with which the songs were recorded.

The Smart Shuffle Feature may not work well when iTunes has problems identifying the same artists with slightly different labeling, i.e. The Smashing Pumpkins vs. Smashing Pumpkins. Also, the options are not clearly labeled, i.e. the groupings option would probably not mean much to someone new to the program.

Storyboard – Using iTunes advanced functionality to delete duplicate songs (by Michael Toulouse)

Step 1. Open iTunes
Step 2. Navigate to View > Show Duplicates

A list of files that are apparently duplicated between the music folders of your library appears.



Step 3. To delete a duplicate file right click on the name of the file, highlight delete, and click on it.

Step 4. A window pops up asking if you would like to remove the selected songs from your iTunes library. Selecting yes removes the song(s) from the playlist.

The song is removed

Notes:
Careful scrutiny of the first two files selected at the beginning reveals that although both contained the artist name, track name, and album information, the song length was not the same meaning these were in fact two different files. It is very possible for two files to share the same file information but be two different songs, for example, in the case of both a live and recorded version of a song. The user may inadvertently delete one of the files from his or her play list without meaning to because of this.

The confirmation window that appears in step 4 is deceptive because it does not delete the file from your computer but merely removes the duplicate reference from your play list. A user attempting to free up hard drive space may be unaware of this as it is not made readily apparent nor is the option for permanent removal presented. Adding to this, if the box marked “Do not ask me again” is selected from this step the user on subsequent tries will not be made aware that they are only deleting the file reference.

Once the duplicate file reference is deleted it is not easy to undue and the file must be readded manually if the user realizes he or she made an error.

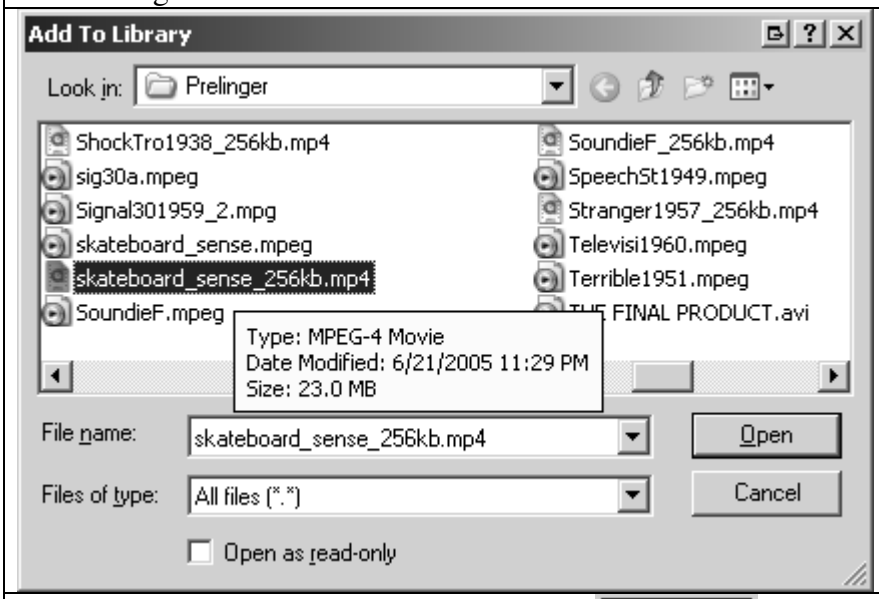
Storyboard analysis by Michael J. Wise – Using iTunes to view a video saved on the user's hard drive and to control playback to move to a specific section of video. Afterwards, they no longer want the video in their library and decide to delete it.

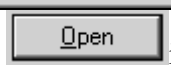
1. Open iTunes
2. Go to Movies library section. (this could also take place between steps 3 and 4)

3. Navigate to "Add File to Library"

Go to "File" > "Add File to Library"

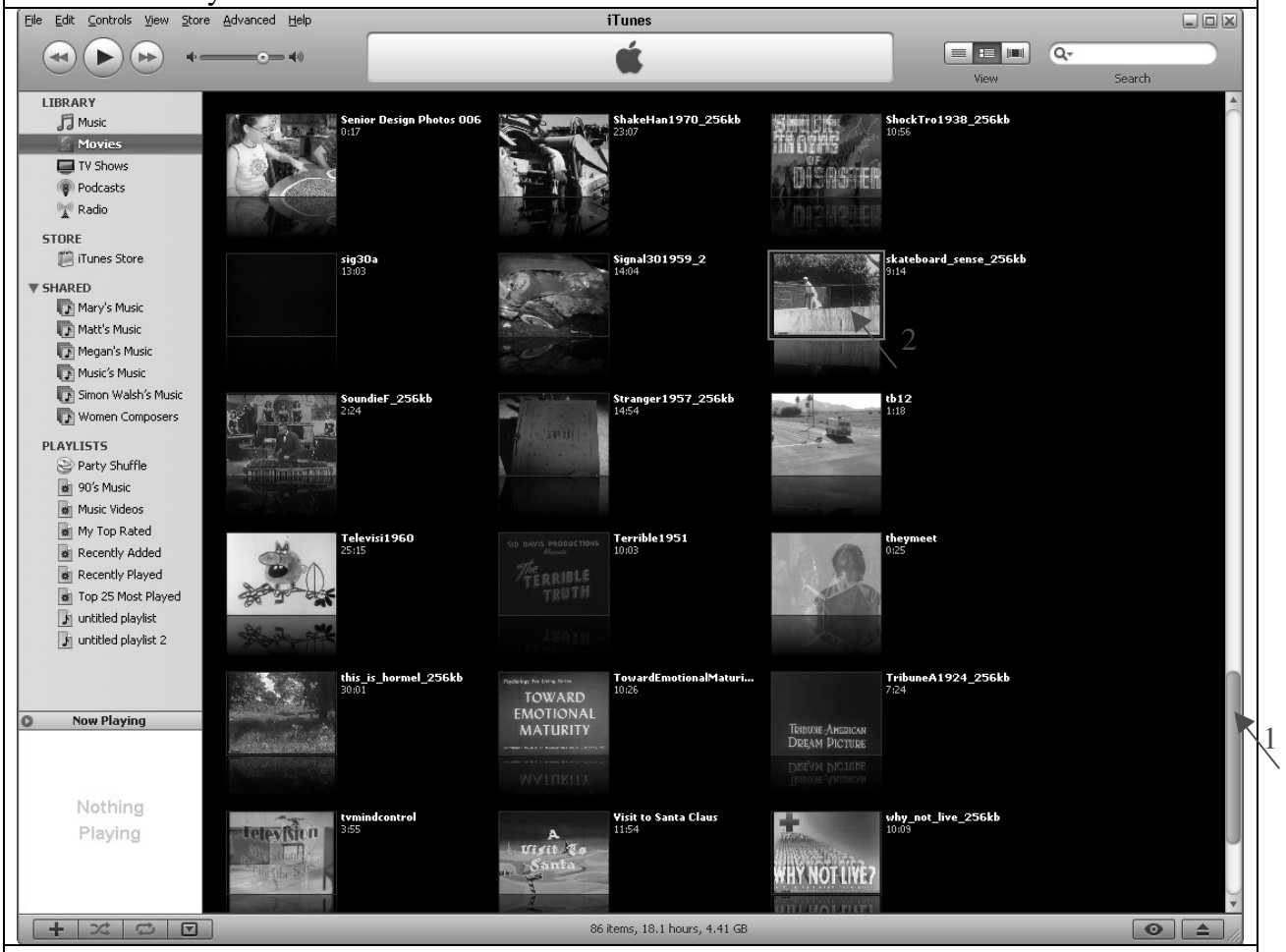
4. Selecting a file….



is clicked on, and 

is then pressed to add it to the library.

5. Movie library list



Navigate to the movies library section and scroll through list of videos (default organization is descending alphabetic order). Doubleclick to start the video, which opens it in a separate window and starts playing the video.

6. Maximizing the video

The user then clicks the  to switch the video window to full screen. Note that the black status bar doesn't appear until the mouse is moved.

7. Selecting relevant position in video

The user then clicks and holds onto the diamond position bar (

) and slides it to the desired location in the video. The video automatically pauses and shows stills from the applicable location of the video that the slider is currently on and resumes playing immediately when the mouse button is released.

8. Moving play control bar

Since information of interest may be blocked at bottom, user moves playback control bar into different location to see more of the window. To do this, the bar must be clicked on in a non-control position and dragged to a different location in the window.

The control bar only disappears if the mouse pointer is not in the control bar area and there is no mouse movement for several seconds (about 4). There is no way to close the playback control directly. Also, single clicking on the video itself will pause it.

After finishing viewing the section of video of interest, the user presses the  to reduce the video back to a windowed state, and then  can be pressed to close and pause the video.

9. Deleting video

The user can then right click and delete the video from the library.

Note that this action does not delete the file from the computer.

Search System storyboard analysis by Voratima

1. Search for songs on the computer
2. Search for the song on iTune Music Store

1. Search for songs on the computer

1.1 Browsing

1.1.1 Browse through view

Users select\ one of the following iTune views: List view, Grouped with artwork view, or Cover Browser view. Then they scroll up and down or left to right to look for the song.

Input:

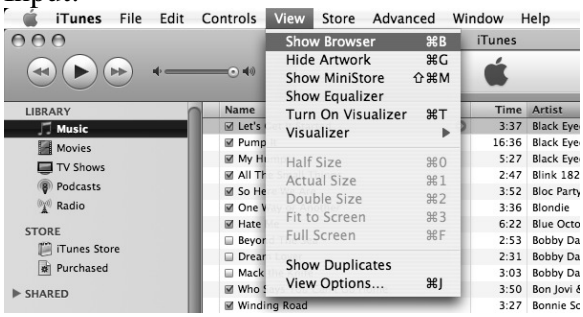
Potential Feedback:

List view Grouped with artwork view Cover browser view

1.1.2 Browse through Browser

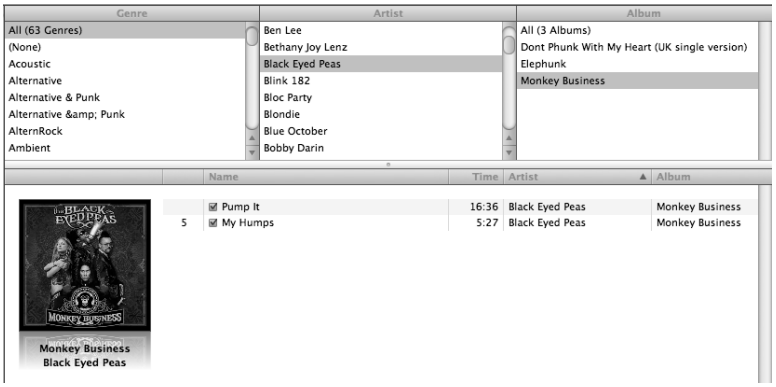
Step 1: To enable Brower, users go to View> Show Browser. This feature is limited to List view and Group with artwork view only.

Input:



Potential Feedback:

Step 2: Then users narrow down the list by select a specific genre, artist, or album. For instance, to view a list of song by Black Eyed Peas, album Monkey Business, users should select Black Eyed Peas > Monkey Business.



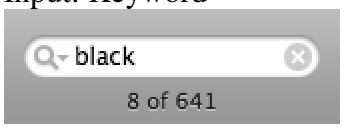
← Input: Genre/Artist/Album

← Feedback: Narrower list of songs.

1.2 Searching

Users enter the keyword in the search box. iTune then displays only a list of songs whose metadata contain the search term.

Input: Keyword



Feedback: A list of matched songs and number of song matched (8 of 641)

Assuming that the song the user is looking for in the computer, if iTune returns no result, there are two possibilities. First, the song is not on the current playlist. The user has to select another playlist and retypes the keyword in order to perform the same search. Second, there is a typo in the search term. iTune does not provide action hint on what the user should do next or what has he/she done wrong.

2. Search for songs on iTune Music Store
   To go to iTunes Store, the user has to select iTunes Store on the left menu.

2.1 Browse
   Browse forces the user to select Genre, Artist, and Album in order. He must know which Genre to start with, otherwise

2.2 Search
   iTune Power Search allows users to search by Artist, Composer, Song, Album, or Genre. The result is a list of songs as well as popular albums, artists, or music video.

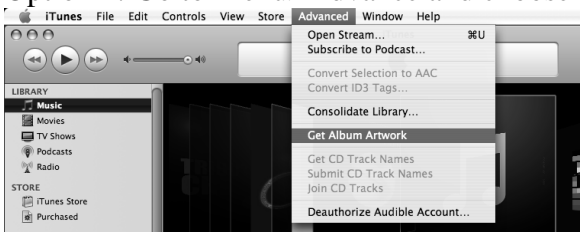
← Input: Artist/Composer/Song/ Album/Genre

← Feedback: a list of matched songs

If the search returns no match, iTune displays an error message that said, “Your search did not match any results. Browse to see all music in the iTunes Store. Request music you can’t find.”

3. Get Album Artwork
   There are two ways the user can get an album artwork.
   Option 1: Right click on a song and choose ‘Get Album Artwork.’

Option 2: Go to Menu>Advance and choose Get Album Artwork.



Feedback: A feedback for this process is not always available. Sometime there is a progress bar indicating the progress; sometime there is not. It is difficult to tell in which circumstance iTune will provide the progress bar. If iTune found the album cover, a blank album cover,  , will be replaced by the actual cover such as  . However, there is no feedback when iTune cannot find the cover.

Importing Music into iTunes and Changing Metadata Storyboard Analysis by Amy Grude

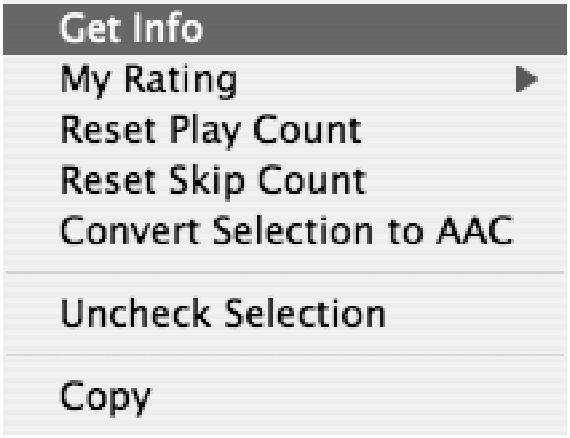
1. Insert CD to be added to library.
   The CD will show up with the songs in iTunes, and a box will pop up asking if you would like to import the CD automatically.

2. Select “Yes”, and iTunes will begin to import the songs.

3. To select only certain songs for import, the user has to select “no” when asked if they want to import. They then click boxes before songs to uncheck them.

4. Click “Import CD”

5. The selected songs are imported. To change the metadata related to the song, right click it and select "Get Info"



6. Select the “Info” Tab and type desired data into data boxes.

7. Click “OK”

Heuristic Evaluations for Nick Senske's Storyboard: Connecting and Disconnecting an iPod to iTunes (Pages B1-B3)

Usability problems in Connecting iPod (Nick storyboard) by Voratima

Step 1: Connecting iPod

Principle violated	Violation Description
Consistency and standards	Users cannot drag and drop files to iPod like the commonly used convention for thumb drive or flash drive.
Visibility of system status	iPod icon first appears on the desktop, available for access, but then suddenly disappears without feedback.

Step 2: Default hardware icon

Principle violated	Violation Description
Consistency and standards	iPod icon is not consistent between PC and Mac. Icon for iPod on PC looks like an external hard. However, icon for iPod on Mac looks like iPod.

Step 3: Do not disconnect

Principle violated	Violation Description
Help users recognize, diagnose, and recover from errors	"Do not disconnect" should be more informative. An alternative message can be: "Synchronizing… Do not disconnect."

Step 4: Sync Music

Principle violated	Violation Description
User control and freedom	Uncheck "Sync music" will erase all songs in iPod. There is no warning message. Users cannot undo their action.

Step 5: Eject iPod

Principle violated	Violation Description
Consistency and standards	Similarly looking eject icons perform different tasks.

Heuristic Evaluations for Ryan Rindler's Storyboard: Inputting general digital music files, folders, and playlists to iTunes (Pages B4-B8)

Usability problems in Import Playlist (Ryan storyboard) by Voratima

Step 1: Importing music or movie

Principle violated	**Violation Description**
Visibility of system status	If the music is playing, there is no feedback for importing files.
Help users recognize, diagnose, and recover from errors	Conversion warning message is not clear whether iTune will alter the file or if the file can still be opened in the old format.
Match between system and the real world User control and freedom	Users cannot uncheck all the songs in a playlist. They have to uncheck the song one by one. Suppose a user has SmartPlaylist that contains only alternative songs namely myAlternative. He cannot simply disable myAlternative to disable all the song on that list.

Ryan Rindler's Usability problems in music file upload / playlist generation by Amy

Adding music files from other sources

Principle violated	**Violation Description**
Minimize information access costs	Unless you are very familiar with iTunes, it's hard to know you must go to "File → add to library" to add from within itunes
User control and freedom	There is no indication that drag and drop is possible
Strive for consistence	The file menu also has an "import" command, which can be confused with "Add File" since import normally means "add something to this system"

Making a playlist

Principle violated	**Violation Description**
Support Error prevention and management	Can add the same file multiple times with no indication from the system.
Visibility of system status	Error messages about incompatible files doesn't inform user of what will happen to original file.
Match between system and the real world	Integrity of folders added is not preserved

Managing playlists

Principle violated	**Violation Description**
User control and freedom	Playlists can only be listed alphabetically with no structure
Support internal locus of control	Converting files blocks the addition of further files
User control and freedom, Match between system and real world.	No way to categorize playlists for uses other than listening to a list of music, for example, setting aside a group of songs for a project or video.

Heuristic Evaluations for Michael Toulouse's Storyboards: Sound Check, Crossfading, Deleting Duplicates (Pages B9-B13)

Ryan Rindler
Usability problems in sound check/shuffle/crossfade feature (Toulouse)

Sound Check

Principle violated	Violation Description
Consistency	The soundcheck and other functions are buried under a preferences/playback menu
User Feedback	Because all songs naturally have varying volume levels, you don't know when the sound check is actually "improving" your experience. There is no way to set up a comparison.

Smart Shuffle

Principle violated	Violation Description
Consistency and Standards	This feature relies on knowing a lot about the iTunes labeling/grouping system. Novice users may be confused.

Crossfade

Principle violated	Violation Description
Strive for Consistency	Like soundcheck, music of varying volumes and structures may not require the same settings.
User Control and Freedom	In cases that soundcheck settings should be drastically altered (dance music vs. spoken word) there is no easy click to turn crossfade on/off.

Storyboard analysis – Deleting duplicates evaluated by Michael Toulouse

Step 1

Principle violated	Violation Description
Recognition rather than recall	The user must remember where the menu location is
Help and documentation	This option is not easily found

Step 2

Principle violated	Violation Description
Visibility of system status	It is not clear at all that you are in the duplicates mode. The library looks almost unchanged provided enough duplicates are found.
User control and freedom	It is not clear how to go back to the normal mode
Help and documentation	Once duplicates are found it is unclear what to do with them

Step 3

Principle violated	Violation Description
Recognition rather than recall	The user must remember to right-click in order to delete the file.
Visibility of system status	It is unclear if the files are truly duplicates or not
Consistency and standards	It is not clear if a file with the same name but different lengths is the same song file or not

Step 4

Principle violated	Violation Description
Consistency and standards	Does delete mean delete the file or just delete it from the iTunes playlist.
User control and freedom	It is unclear how to bring the file back once deleted.
Help users recognize, diagnose, and recover from errors	The do not ask me again box once checked causes the file to be deleted from the playlist without any sort of confirmation

Heuristic Evaluations for Michael Wise's Storyboard: Video/Movie Viewing (Pages B14-B20)

Usability problems in movie addition/playback (MJW storyboard) by Michael J. Wise

Step 3

Principle violated	Violation Description
Recognition rather than recall	User must know to go to file menu and know to add a file to the library in order for it to appear. No common "open file" selection and no way to directly open a file from the hard drive.
Multiple Resources	No input methods other than mouse and keyboard are supported

Step 4

Principle violated	Violation Description
Consistency	File dialog uses "open" terminology yet it does not have that functionality.
Recognition rather than recall	Default file filter in file dialog is inexplicably "all files" which causes all files to be listed to potentially add. This is incorrect behavior since iTunes only supports certain media files and certainly not all files.
Recognition rather than recall	File open dialog is generic and does not support a structured file search, only file browsing

Step 5

Principle violated	Violation Description
Support internal locus of control	File just added is not scrolled to automatically, nor is the correct part of the library automatically chosen. (If a movie file is chosen to be added while in the music library, the user remains in the music library after adding it. User must scan through list to find just added file. Recently added and opened materials should be automatically available for usage.
Consistency	Scroll wheel is supported. Inconsistent with rest of windows interface with scrollbars.
Easy reversal of actions / Consistency	No simple undo capability to undo the add file action. Standard Ctrl-Z functionality is not present.
Design dialogs to yield closure, helping users to recover and diagnose from errors	Unclear that the add file function even does anything…furthermore if a non-media file is added, the dialog silently fails and does not inform user if anything was added. There is no feedback whatsoever from the dialog.

Step 6

Principle violated	Violation Description
Recognition rather than recall	Iconography for maximization is not standard for Windows (or mac?) and is not explained in the interface via any sort of online help.
Internal locus of control/Reduce short term memory load	It is not clear how to control the video as the black on-video status bar does not appear unless the mouse is moved while the video window is active. Video can be controlled with standard music control bar but this causes the video screen to go to the background.
Consistency	Music always is controlled using the bar at the top of the iTunes window, but if this is attempted with a movie one ends up

Step 7

Principle violated	Violation Description
Internal locus of control	There is no way to close the black status bar except by refraining from moving the mouse for several (approximately 4) seconds. There is no way to directly and quickly dismiss the black status bar

Step 8

Principle violated	**Violation Description**
Recognition rather than recall	It is not clear how to move the status bar and no visual cues are given simply to grab onto an "unused" part of it and move it directly
Consistency	If the image is inadvertently clicked on, it will immediately pause. However, clicking on it does NOT unpause it. Double clicking to resume is not an immediately obvious function and is not at all consistent with how other media players operate, notably Windows Media Player.
Consistency	Unlike music files, movie files are resumed for whenever they are last paused if they have been previously partially viewed (and no matter how long it's been since they were last viewed, apparently). This is not how music files behave.
Enable frequent users to use shortcuts	There is no way to close the video from its full screen state without first restoring it to its previous size.

Step 9

Principle violated	**Violation Description**
Consistency	Delete does not actually remove the file from the computer. Delete is usually reserved for when a file is expunged from a computer. (Remove might be a more accurate word?)
Easy reversal of actions	No way to easily undo the action. Must go through the entire add file procedure to get the video back into the library

Storyboard analysis – using iTunes to view a saved Video evaluated by Michael Toulouse

Step 1 through 3

Principle violated	Violation Description
Recognition rather than recall	The user must remember that to play a video file it must first be added to the playlist and that a "file" can be both a video or a song

Step 4

Principle violated	Violation Description
Match between system and the real world	The system displays the files names without stating the type of file that they are. The user is forced to remember that a .mp4 file extension represents a video file

Step 5

Principle violated	Violation Description
Visibility of system status	The system does not highlight the newly added video, user is forced to search for it.
Help users recognize, diagnose, and recover from errors.	If the user adds the wrong video there is no clear way to un-add it.

Step 6

Principle violated	Violation Description
Match between system and the real world	The button to maximize the screen is vague

Step 7

Principle violated	Violation Description
Help and documentation	A user unfamiliar with a video playback bar may be confused here.
Aesthetic and minimalist design	The progress bar is displayed over the video and must be moved requiring further action from the user.

Step 8

Principle violated	**Violation Description**
Visibility of system status	The video, once maximized, takes us the whole users desktop, to play another video the current one must be minimized.
Visibility of system status	The progress bar disappears if the mouse is not on the video anymore

Step 9

Principle violated	**Violation Description**
Match between the system and the real world	Deleting the video does not delete it from the computer, merely from iTunes. This is not made very clear.

Heuristic Evaluations for Voratima's Storyboard: Search System, Music Store, and Album Artwork (Pages B21-B24)

Usability problems in Search, Browse, and Get Album Cover (Voratima storyboard) by Voratima

Step 1.1: Browse Song on the local machine

Principle violated	Violation Description
Matching between the system and the real world Consistency and standards	The Grouped by Artwork view is confusing; users are not certain of the songs' order in this view.

Step 1.2: Search Song on the local machine

Principle violated	Violation Description
Recognition rather than recall	Album cover feature has pros and cons. On one hand it helps users recognize the songs in the album. On the other, users have to know which song is in which album (recall).
Help users recognize, diagnose, and recover from errors	There is no feedback when there is no match found.
User control and freedom	Users do not usually know which playlist they are currently looking at. After searching for a song and realizing that there is no match found, users then glance over to the playlist and discover that they are in the wrong one. They have to retype the search term when switching to another playlist.

Step 2.1: Browse Song on iTunes Store

Principle violated	Violation Description
Recognition rather than recall	When browsing for a song, users must recall the song's genre, artist, and album consecutively. The browsing order cannot switch around. For example, if users cannot come up with the genre, it is not possible to reach the song they are looking or even see a list of song.

Step 2.2: Search Song on iTunes Store

Principle violated	**Violation Description**
Help users recognize, diagnose, and recover from errors	The feedback is vague and could be improved buy suggesting, “do you mean to search for x.”
Match between system and the real world	Users usually search for artist, song, or album first. Then they might narrow the search by genre or composer. The order of search boxes on iTuens Store does not consistent with those of users. They are ordered by artist, composer, song, album, and genre.

Step 3: Get Album Cover

Principle violated	**Violation Description**
Visibility of System	The progress bar indicating downloading progress is not always available. There is no feedback whether the album cover is available. When the album cover is loaded, iTunes provides poor feedback by sliding the cover over to the other cover. As a result, users have to scroll back to check if the cover is found.
User control and freedom	Users cannot cancel get album cover request.

Search + album art procedures - Voratima storyboard by Amy

Step 1.1.2

Principle violated	Violation Description
Recognition rather than Recall / Match between system and real world.	There is no indication that the browse feature is even available, unless you know to look for it. The browser is located under the "view" menu, even though "view" conveys the passive act of looking at something. Browsing seems like it should fit in a location that implies some sort of action.

Step 1.2

Principle violated	Violation Description
Match between system and real world	The browser allows users to browse only by a certain genre, artist, or song name. However, many songs are missing metadata, and in real life, people follow paths from one thing to another while browsing. This is not possible in the iTunes browser.

Step 2.1

Principle violated	Violation Description
Visibility of system status	When searching, it's often not clear that you are within a certain playlist and only searching that playlist – the user can easily be searching from the wrong place and not know it
Help users recognize, diagnose, and recover from errors	When you search from the wrong place and are unsuccessful, there is no way to tell why you are unsuccessful unless it occurs to you that you are searching from the wrong place.
Permit easy reversal of actions	When you search from an unintended playlist and switch back to the entire library or a different playlist, the search terms you initially entered disappear

Step 2.2

Principle violated	Violation Description
User control and freedom	Users are only able to browse by genre

Step 3

Principle violated	**Violation Description**
Support error prevention and management	If iTunes has some kind of error or can't find artwork,
Consistency and standards	iTunes feedback on this process and what is happening is inconsistent

Usability problems (Search + album art procedures - Voratima storyboard) evaluated by Michael Wise

Step 1.1.2

Principle violated	**Violation Description**
Recognition rather than recall	Not obvious how to enable search browser
Permit easy reversal of action	No history function to recreate or recall previous search narrowings

Step 1.2

Principle violated	**Violation Description**
Visibility of system status	No feedback that only the current playlist is being searched. Search dialog seems "global" up in the corner yet is only apparently used on current playlist/library.
Support error prevention and management	Search is exact only, no "fuzzy" searching to find very similar items.
Reduce short-term memory load	Search box contents are erased as playlist or library location is changed, necessitating recall and re-entry if the same search is to be executed across playlists.
Reduce short-term memory load	No history function for searches

Step 2.1

Principle violated	**Violation Description**
Internal locus of control	User is not allowed flexibility to search for genre, artist, album in any desired order.
Consistency	Order choices of genre, artist, album are required to be made are not consistent with the browser in the library and playlists

Step 2.2

Principle violated	**Violation Description**
Consistency	Several controls available here that are not available in other search modes, including browser-like forward and back buttons

Step 3

Principle violated	**Violation Description**
Design dialogs to yield closure	If you are not signed in, you are told to sign in or create an iTunes Store account, but software does not offer to simply take you there directly
Offer informative feedback	No feedback when album art cannot be found

Heuristic Evaluations for Amy Grude's Storyboard: Importing Music into iTunes and Changing Metadata (Pages B25-B30)

Amy's Importing CDs and changing metadata – By Amy

Step 1

Principle violated	Violation Description
Support/Internal locus of control.	The CD and its songs automatically show up in the window, replacing whatever the user happened to be working on. This is great if the user wanted to deal with the CD then, but not if the user didn't intend to work with the CD in iTunes or was using iTunes for something else.

Step 2

Principle violated	Violation Description
Support Internal locus of control	Selecting "yes" you would like to import songs immediately begins to import all songs, without giving the option of only importing several songs.
Permit easy reversal of actions	It is not clear how to stop importing songs (click the x on the status bar)

Step 3

Principle violated	Violation Description
Support Internal locus of control / offer informative feedback	To select only certain songs, the user has to select "no" when asked about importing, which is counterintuitive.

Step 5

Principle violated	Violation Description
Minimize information access cost	To change metadata, the user has to right click and select "get info". This label does not indicate that this is a place were you can change things rather than just view.

Usability problems (Importing CDs - Amy Grude storyboard) evaluated by Michael Wise

Step 1

Principle violated	**Violation Description**
Internal locus of control	CD is automatically brought up in favor of what the user was currently doing – perhaps not what they wanted to do? User does not initiate the action.
Recognition rather than recall	"Import" term is not clear as to its function. Not clear it is copying music to hard drive.

Step 3

Principle violated	**Violation Description**
Recognition rather than recall	Checkbox functionality is not clear – it controls playlist and also chooses what files will be imported if "Import CD" is checked. "Import CD" could be more descriptive – implies whole CD will be imported no matter what.

Step 5

Principle violated	**Violation Description**
Recognition rather than recall	"Get Info" is not indicative what info is being brought up and that it can even be changed.

Step 6

Principle violated	**Violation Description**
Recognition rather than recall	"Grouping" field is ambiguous
Reduce short-term memory load	No provision for selecting already-present names, artists, etc. from files in the library.

Storyboard analysis – Deleting duplicates evaluated by Michael Toulouse

Step 1

Principle violated	Violation Description
Help and documentation	It is unclear how to only import a part of the cd at this point
User controls and feedback	Once the do not ask me again box is checked cd's are either imported or not imported automatically, the user may not be aware of this

Step 2

Principle violated	Violation Description
Help and documentation	Selecting yes did not give you the option of picking which songs to import
Visibility of the system status	The user is not informed as to where the files are going and if in the library view may not see the new files appear

Step 3

Principle violated	Violation Description
Recognition rather than recall	It is unclear what the check boxes next to the song titles mean. They could very well perform multiple functions.
Visibility of system status	The system does not seem to inform the user that the import feature for individual songs is readily accessible here

Step 5

Principle violated	Violation Description
User controls and feedback/visibility of the system status	The user is not really sure if in the wrong mode where the imported songs went
Help users recognize, diagnose, and recover from errors	It is unclear how to un-import songs
Recognition rather than recall	The user must remember that to modify songs a right click is required

Step 6

Principle violated	**Violation Description**
Help and documentation	Is not apparently available to describe what the different options mean
Match between system and the real world	The system uses words that a novice user might not be familiar with

Name: Jonathan

Gender: Male
Age: 32
Occupation: Freelancing graphic design artist
Education: Bachelor's degree
Computer Experience: Extensive on
Mac
Computer Owned: Apple Power Mac G5 Tower
Internet Connection: DSL 512k
Country/Language: US, English
Hobbies: Tennis

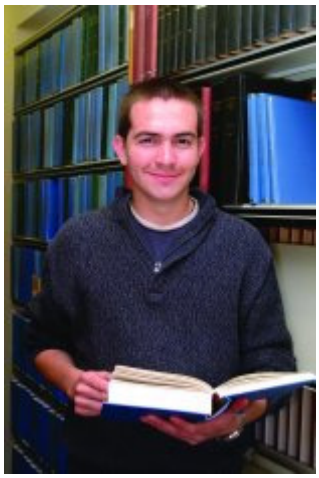
Persona:

Trevor graduated with a BA in Graphic Design approximately 10 years ago and after a short stint with a major consulting firm, he decided to start his own business. His customers are mostly small, local businesses at this point although he did recently sign a contract to provide graphic design services for a regional office of a Fortune 500 company. He gets to set his own hours and often finds himself starting his day later and ending it even later.

Given that his work is visually intensive, Trevor listens to his extensive iTunes library as he works. He has ripped his entire CD collection to his hard drive and also owns an iPod Nano. His mood and musical desires change frequently throughout the day and he enjoys the browsing ability iTunes affords.

In his free time, he is part of a tennis club. He has had a steady circle of friends since he graduated from college and they gather at his place once or twice a month to play cards or watch a movie and catch up.

Name: Daveryn



Gender: Male
Age: 22
Occupation: Student (Pre-Law)
Education: Diploma
Computer Experience: Familiarity with common PC functions
Computer Owned: Dell Inspiron E1705 Notebook
Internet Connection: Broadband (wireless)
Country/Language: US, English/Spanish
Hobbies: Swimming, Debate

Persona:

Daveryn is currently in his final year at the University of Virginia, majoring in Political Science and History. He plans to attend law school next year, and eventually practice Constitutional Law. Throughout his coursework and during his two internships at a local law firm, Daveryn has extensively used many common software applications (word processing, research tools, spreadsheets). However, like most students, he also takes his notebook with him to classes, the library, and coffee shops, and has explored more of what his computer has to offer.

Also like most students, Daveryn listens to music almost everywhere. He brought his entire CD collection to college, and has uploaded a handful of them to his computer. Although he downloaded iTunes, he prefers Winamp for everyday music listening. He still finds that (to a limited degree) he can share and obtain new music, but is thinking about switching to iTunes if he buys an iPod sometime soon.

Daveryn is excited to finish his studies at Virginia, but is worried that he and his girlfriend will not get into the same Graduate Schools (she is studying Education). In fact, most of Daveryn's immediate and extended family live in northern Virginia, and it is likely that he will be moving away from them for the next few years. Eventually, he would like to establish his career in the Washington, D.C. area to be closer to home.

Name: Tammy

Gender: Female
Age: 35
Occupation: Writer
Education: Bachelor's degree in Literature
Computer Experience: Some Mac
Computer Owned: iBook 12.1" G3
Status: Recently Single
iPod: Generation 5.5, 30 GB

Persona:

Tammy is an aspiring writer with a promising career. She works at a publishing company in Chicago. Her column is biweekly and it is about women and fashion.

Tam uses her babe, iBook, as a secret weapon when it comes to writing. Because of her flexible schedule, she does not have to come to her office everyday and often bring her column with her to a local coffee shop or her apartment. Other than using it to write, Tam uses her babe to send and receive emails and chat with her friends via Instant Messaging. As an expert in fashion, she has minuscule knowledge of what her computer can do.

On her 35th birthday, her big sister, Jane, gave her an iPod. Jane believes iPod would be perfect for her sister in many ways. It has voice recorder capabilities—perfect for writer. Also, Tam lives 5 hours away from her hometown; she drives home to visit her family at least once a month. Knowing her sister is a clumsy driver, Jane does not want Tam to change the CD while driving. In her opinion, iPod serves this purpose perfectly; it plays hours of music and allows her sister to keep her eyes on the rode.

Knowing iPod has become part of the culture and fashion, Tam is very excided to put it into good use. As a dedicated jazz fan, Tam has piles and pile of jazz CDs in her apartment. She can see the benefit of using the iPod to organize her music collection.
iTune, however, is something she is skeptical about. Since it is the only way she can download music to iPod, she has to learn how to use it.