User Interface Design for a Content-aware Mobile Multimedia Application: An Iterative Approach

Zahid Hussain, Martin Lechner, Harald Milchram, Sara Shahzad, Wolfgang Slany, Martin Umgeher, Thomas Vl, Peter Wolkerstorfer
Graz University of Technology, Institute for Software Technology
m3@ist.tugraz.at, www.ist.tugraz.at
CURE - Center for Usability Research & Engineering
wolkerstorfer@cure.at, www.cure.at

Abstract:
Mobile phones have become full-featured mobile computers. Applications providing good user experience and taking full advantage of the increasing capabilities of mobile phones are still rare. One such application is audio and video on mobile phones which is expected to become a killer application in the near future. A lot of valuable audio and video content is hidden in archives of content providers which waits to be explored and utilized. We are developing an application that enables a user to perform content-based search for any audio and video content in a large database and play it on a mobile phone. It allows users to consume multimedia content virtually anywhere, at any time, requiring no additional software on modern mobile phones. Our novel approach to application development focuses on the adoption of agile software development methodologies and usage-centered design, emphasizing iterative user-interface development and usability engineering. The feedback from the usability engineers is incorporated continuously into the system. Thus, the application evolves according to the needs of the end user, providing maximized usability and customer satisfaction. In this unique context the key success factor is user acceptance. For this reason we choose Extreme Programming as methodology, enhancing it with usage-centered design aspects involving usability engineers and non-technical users.

Keywords: Usability, Iterative User Interface Design, Mobile Application, Multimedia, Extreme Programming

1) The research herein is partially conducted within the competence network Softnet Austria (www.softnet.at) and funded by the Austrian Federal Ministry of Economics (bm:wa), the province of Styria, the Steirische Wirtschaftsförderungsgesellschaft mbH. (SFG), and the city of Vienna in terms of the center for innovation and technology (ZIT).
1 Introduction

Mobile computing is leading a revolution. Our lives are changing at a pace never experienced before in human history. A wide variety of applications for mobile phones is available at the moment. Still, there are not so many full-featured applications which utilize the available bandwidth and are accepted by the users.

Studies show that multimedia – Audio and Video (AV) – consumption is on the edge to become one of the next killer applications for mobile devices [?]. User behavior in consuming AV is changing. Traditional broadcasting is losing more and more audience because online and mobile AV intrudes heavily into this area. A recent report states that 43% of Britons who watch video regularly from the Internet or on a mobile device are now watching less TV than before [?]. Clearly, it is in the interest of broadcasting companies to adapt to these changes in user behavior and invest in these new technologies. For these companies, one of the major advantages of mobile phones compared to other devices is that they can charge for their services easily and directly, as the existing infrastructure can be reused. Additionally, the possibility to place advertisements for specific user groups is a huge benefit. At the same time, customers are given the flexibility to access rich multimedia content from anywhere, at any time.

The major problem for an average user is the combination of the overwhelming amount of multimedia content available and unsatisfactory user-interfaces for accessing it. Usability is the key success factor for such applications.

For this reason we are developing an application that enables a user to perform content-based search for AV content and play it on a mobile phone. This content includes radio and TV archive material, like documentaries or other recordings of historical, political and cultural importance, discussion programs, movies, music videos, audio books, and music. The AV content will be stored in a database containing transcribed speech from the clips as well as additional metadata, such as titles or a summary, where available. The media delivery will be based on standard web technology. This will enable people to use this service with almost any modern mobile phone. Furthermore the application addresses not only the emerging functional and cognitive needs of the users but also the objectives of the content providers.

The application is designed keeping in mind the social interaction aspects of users. The system provides different community-building features to encourage interaction amongst them. The aim is to build a community platform for mobile phone users where they can share their views and interests about AV content provided by the system. The feedback from the community will reflect the current trend of multimedia consumption as well. Covering these aspects in detail goes beyond the scope of this paper.
One of the research goals of this project is to apply usability test procedures for mass-market applications on mobile phones. At present, usability testing for mobile phones is cumbersome and too expensive for small and medium sized enterprises. Another objective of this project is to automate some parts of the usability testing procedures and provide a testbed for effective and efficient mobile usability testing. Special emphasis is placed on the adoption of agile software development methodologies, in particular Extreme Programming (XP), for mobile phones and their user-interfaces.

This paper presents an overview of different usage scenarios of the system. The usability engineering process is described which is applied to the iterative UI development cycle. The next section defines various user-based recommendation approaches. Finally, the architecture of the system is given.

2 Usage Scenarios

In daily life, many people are at work or school at day time. They have no time to watch TV or listen to radio programs, as the schedule is set by the broadcasting companies. Time-delayed, played-back, and individually-delivered AV on mobile phones provides a new platform taking radio and TV into the street, car, public transport, waiting room, park, and virtually any other location. This type of application creates additional audience eager to access multimedia content during new prime times set by themselves, that is, in their commuting periods or other idle times. Also, a market survey shows that consumers are interested in using this technology and are ready to pay a realistic price for these services [?]. The basic idea of such a system can be illustrated by the following sample usage scenarios.

2.1 TV Archive for Subway Riders

- A commuter in the subway searches for “Fernando Alonso”.

- The system matches each word of the textual search query with the positions it occurs in each AV clip.

- The system presents a list of clips in which the name “Fernando Alonso” has been mentioned, sorted by temporal occurrence and relevance based on content, e.g., how often the name was mentioned.

- The user selects one of the presented entries.

- The system’s media server delivers the selected clip to the user’s mobile phone.
2.2 Radio Archive for Car Drivers

- A user listens to the last sentences of a radio broadcast about the “European Constitution”. The user still has to travel with the car for some time and therefore searches for the keyword “European Constitution”.

- The system lists a number of related news items, interviews, and documentaries.

- The user selects the desired topic.

- The handsfree set of the mobile phone plays back the selected material through the car’s stereo.

2.3 Media Recommendations for Users

- A user wants to consume some AV content but has nothing particular in mind.

- The user asks the system for recommendations.

- The system generates recommendations based on the user’s stored preferences and on other users’ recent behavior. A short description of each item is also provided.

- The user selects an item and plays it on the mobile phone.

2.4 Media Rating by Users

- A user listens to some AV content.

- The user rates the clip.

- The user rating is stored in the system and used for further recommendations.

If the user interrupts the media stream, in all scenarios it is possible to resume at the previous location at any time, even weeks later. This feature is unavailable with regular broadcasting or streaming systems. The user of this system has more flexibility for consuming the AV content. This is particularly important because of the short continuous viewing or listening periods. For example, while commuting, interruptions and (possibly much) later resumptions will be the regular case.

Such behavior is rather uncommon for AV consumption so far, especially for viewing video. But it is not so much different from the way a book is read, having breaks between reading periods. Thus, it seems plausible that users will be willing to switch to this new way of
listening and viewing AV content with interruptions as it brings them the convenience of being able to decide what to consume in a just-in-time way, independent of place and time.

3 Usability

User-interface design determines the success or failure of almost any application. Massive AV consumption on mobile phones will be accepted only if users can easily find what they are searching for. But search on a mobile phone presents unique challenges as compared to a PC. The inherent interface limitations of mobile phones strongly constrain the choices of user-interface and interaction design. Special attention has to be paid to the constraints of small screens, possibly unfavorable lighting conditions, and limited text input capabilities.

We propose an iterative and usage-centered approach to user-interface design and system development in order to solve the stated problems.

3.1 Iterative User-Interface Design

Usability is evaluated in small iterative steps to gain insight into whether the users’ functional and cognitive requirements are met. User-interface prototypes of the system are developed and tested throughout the development process. As a result the fidelity of the prototypes increases and evolves.

![Figure 1: Iterative UI design workflow.](image)

The workflow presented in figure illustrates the iterative design approach. The process starts with the creation of user stories by the customer or the product manager who acts as
a representative of the customer. Developers create different paper mock-ups to collect and present ideas. A final mock-up is derived, serving as the basis for further development. The benefit of using paper mock-ups for the interaction design is that they can be designed and modified quickly. Because of that, the feedback given by the usability engineers and the users can be easily incorporated. An additional advantage is that it is easier for users to criticize simple and rough mock-ups compared to ones which look neat and perfect from the graphic design perspective [?]. For simple interaction designs, a paper mock-up suffices as a basis for further discussions and the implementation. In more complex cases, an additional HTML mock-up is created based on the final paper mock-up.

The approach combines the quick feedback-and-change cycle of hand-drawn paper mock-ups with the more time-consuming process of computer-based prototypes. Paper mock-ups are used to get the basic concepts right while HTML mock-ups are used for a more detailed view.

The designs are examined by usability engineers and tested by non-technical users. The feedback from the usability engineers as well as from the users is taken as input for further refinements of the design. Also, the results are incorporated into automated tests which are used as an executable specification for the actual implementation. This feedback-and-change cycle provides insights into whether the user-interface design is meeting usability criteria.

For the actual user tests it is important to choose representative test users from different age groups, bearing in mind the targeted customers for the proposed service. These tests are conducted only after incorporating the feedback on the paper as well as HTML mock-up from usability engineers. Therefore, the expensive part of involving real users can be done more effectively.

### 3.2 An Iterative Design Example

![Figure 2: From paper mock-up to mobile: the first search-results screen.](image)

(a) Paper mock-up.  
(b) Application on mobile.

For the paper mock-up in figure ?? the usability engineer raised the following issues:

- Missing strategy for displaying larger result sets (balance between pagination and...
scrolling).

- Missing feedback mechanism to highlight the pointed-to item (especially needed in unfavorable lighting conditions).

- Undefined application behavior after playback of the clip ends (e.g., no return option specified).

Figure 3: An additional HTML mock-up: a refactored search-results screen.

Figure ?? shows the mock-ups of an improved version of the same feature. Additional to the process depicted in figure ??, in this scenario an HTML mock-up was created after the paper mock-up. The design was derived from the following user scenario (a so-called user story in XP [?]):

Search results presented to the user should contain clip-related information which can aid the user in choosing the clip. Also the context in which the keyword was found as well as the number of search results should be visible. Furthermore it should be possible to start a new search immediately.

It can be seen that two issues from the previous feedback, namely pagination/scrolling and item highlighting, have been addressed in the refactored design.

3.3 Usage-Centered Application Design

The user-interface development cannot be separated from the development of the underlying application. Intended user interactions strongly influence the internal structure and functionality of the system [?]. A big issue in mobile user-interface practice is that current approaches are not sufficient for mobile phones [?]. Therefore, another focus is placed on usage-centered application design.

The design process and user tests provide feedback about the user interface which will be used for the system’s functional requirements. It reveals the users’ mental model of how they expect the system to work. The assessment of each feature from the users’ perspectives
influences the whole development process of the application and addresses the problem that conversation only with the stakeholders is not enough [?]. As the application development is done in short iterations, the developers are able to refactor the system continuously according to the feedback given. Hence, the system evolves according to the needs of the end user and the specifications derived from actual usage.

4 User-Based Recommendations

Web-based companies already use recommendation systems with great success. Amazon, for example, has millions of customers. Seeing the benefits of recommendations, Amazon has developed its own technique called “item-to-item collaborative filtering”. Their customers regularly take advantage of these recommendation facilities when making their purchases [?].

The personalized approach of our system makes it possible to implement user-based recommendations. The unique identification of a user is necessary for accounting purposes, implying that a user profile has to be managed by the system. This profile will be augmented with additional data which is used for recommendations to the same and to other users. The data is collected by means of two information acquiring models, the interactive model and the behavior-based model.

4.1 Interactive Model

The interactive model is based on user ratings. After users finish consuming an item, they are able to rate it according to their liking and preferences. Information about clips that users consumed and their respective ratings are stored in their individual profiles. The rating of a clip in each user’s profile affects the overall rating of the clip in the database. The individual ratings are still traceable. For more personalized recommendations, ratings of similar user groups can be combined.

4.2 Behavior-Based Model

The behavior-based model is applied by collecting usage data. Information about the clips consumed and the duration of the consumption is stored in the users’ profiles. This is used for user-specific recommendations. If many users stop the same clip after a short time, this clip is most likely not very interesting. Of course, this equally depends on the overall playing time. Therefore, a ratio measure is used for clip rating. The system will take into account that users are allowed to stop and resume clips at any time which can influence the measurements. Alternatively, it is possible to consider ratings of a specific user group only, as described in ??.
4.3 Model Combination

These two models are combined when generating recommendations. For the system, user ratings are more important than usage data. However, ratings may not be available for every item. In this case only behavior data is used. Furthermore, the changing preferences of users are taken into account by adding a time-descending weighting factor.

There are different scopes for the rating mechanism. On the one hand all users are considered, and on the other hand just a specific user group is considered. This results in different recommendations. The default recommendation setting can be overridden by user preferences.

4.4 Implications

An attractive feature of the system is the possibility to target advertisements more precisely. This feature is useful for companies wanting to address specific user groups. Additionally, users benefit because they receive only advertisements related to their interests. For example, Google’s Gmail is using this technique for advertising purposes on its popular mail accounts. The large user base of Gmail is a valuable target for business. The advertising is tailored to users’ mail contents. Gmail also offers the possibility to use mobile devices [?]. It is expected that this trend will continue as Google has purchased YouTube. The advertising and search capabilities are, or will soon be, extended also to video content [?].

The feature of collecting additional user data provides continuous feedback, enabling constant improvement of the system. By recording this information, valuable data about how the user is interacting with the system is obtained. This allows to react quickly to new usage patterns and needs as they arise.

5 Architecture

![Figure 4: System Component Diagram](image)
Figure ?? presents a coarse grained overview of the components of the system. The server-side system comprises a database, a media server, and a web server. The client side is represented by mobile phones with multimedia and web browsing capabilities.

5.1 Database

One of the essential parts of this system is the database containing the AV data, the metadata, and the transcribed speech. In the database, AV content can be stored in any standard format and codec. However, because of technical restrictions of the target platform the clips are transcoded to a compatible format (MP4 with H.264) in a preprocessing step. The metadata that is attached to the AV data enables additional search possibilities within the database. The system allows to search in metadata (e.g., title, descriptive text) as well as in the transcribed speech. Furthermore, the transcription data can be used to search for speakers.

The process for generating metadata and transcription text takes an audio or video file as input. It consists of the following steps [?]:

- Speaker change detection to label audio input as speech or non-speech
- Automatic speech recognition to transform audio into text
- Speaker identification to cluster utterances of the same speaker
- Named entity detection to identify, e.g., persons, locations, or organizations
- Topic detection to identify segments of homogeneous topic content

The indexing process produces an XML document containing the spoken text, annotated with speaker ID and time stamp. This XML document is then stored in the database.

5.2 Web Server

The web server manages client requests and hosts the user-interface for browsing and searching in the database. Because of the diversity of available mobile phones, the system is web-based and is accessed through the pre-installed web browsers of mobile phones. Therefore, the challenge of having to deal with many different client devices is alleviated.

5.3 Media Server

The media server is responsible for delivering the actual AV content to the client and supports advanced user interaction. This includes stopping, fast forwarding, skipping parts of the AV
content, resuming later, and replaying a sequence. In this project, the open-source streaming server Darwin [?] is used.

5.4 Mobile Phone

The system does not require a specific client software to be installed on the mobile phone. Instead, any state-of-the-art HTML browser and any video player capable of receiving AV streams can be used.

6 Conclusion

The emerging technologies of delivering rich AV content on mobile phones will result in reducing the number of users for traditional TV and radio broadcasting services. This might compel traditional TV and radio broadcasting companies to become partners in this technology by opening their huge collections of AV content to the public. Today’s consumers are willing to pay a reasonable price for this service [?]. According to current trends, the community of mobile phone application users will grow rapidly. The standards concerning codecs, formats and technical infrastructure required for AV content delivery on mobile phones are already well established. These general trends are in favor of the development of this type of application.

The critical factor for this kind of applications will be user acceptance which depends heavily on the fact that the system suits the users’ needs. To address this issue, in our software engineering process, usability engineers are accompanying the system development team during the whole project life cycle. The engineers provide suggestions that are incorporated continuously into the system. This process is facilitated by the short development iterations and has proven to provide early and valuable feedback. The test-driven development approach allows to convert these findings into a set of automated tests. These tests define the functionality of the system, serve as specifications for the development, and prevent previously discovered usability problems from reappearing. Furthermore, the inclusion of test-users provides additional benefits. This continuous input allows to adjust the system effectively according to the end-users’ needs.