SCRAPECARDS: FLEXIBLE FLASH CARDS FOR UBIQUITOUS MICRO-LEARNING

Roman Brandt and Dejan Kovachev

Advanced Community Informations Systems (ACIS) Group
RWTH Aachen University
Aachen, Germany

Mobile flash-card learning apps proliferate since the mobile platform is ideal for learning on small pieces of information, i.e. micro-learning. Using flash cards for acquiring domain knowledge facilitates memorization by cued recall repetition. However, existing systems impose functional constraints that hinder context-aware and frictionless micro-learning. In this paper, we present our μLearn prototype which addresses the issues of fast creating and seamless consuming of flash cards for ubiquitous micro-learning. The initial results yield to be promising for the chosen approach.

I. Introduction

Life-long learning is a crucial but hard to balance demand in the face of competing demands of work and private life. Many life-long learners simply do not find the time or do not have the resources for intensive or immerse instructions. For this reason, life-long learners need to identify and exploit the breaks spread across different places and throughout the day/week and when other distractions are temporarily halted and attention can be diverted to learning [1]. Yet another challenge is the curating of reusable personalized learning content and tools. Information overload is a much greater threat than information scarcity. There is a plethora of learning and informational content on the Internet, but there is also too much of it and the learners are flooded with information and tool choices. Losing control and focus often lead to non-satisfying learning experiences and may result in stopping all learning activities.

With mobile and cloud eco systems new heterogeneous kinds of informal learning are now available to life-long learners. Nowadays, people use multiple personal computing devices like desktop computers, laptops, smartphones and tablets. These devices are used in different situations, for different reasons and in different periods. The combination of all kind of devices is a solid basis for improving the learning process of every user.

The Web has become the main source for informal life-long learning. Web 2.0 platforms such as Wikipedia, blogs, Q&A sites and online dictionaries are used to abate gaps in personal knowledge. The Web 2.0 platforms, however, often contain overabundant information than the user needs. Scraping some parts of a complete website and saving it, helps the user to collect only the information he/she needs to understand certain topic.

This paper presents μLearn - a prototype system that enables the user to collect pieces of information from web pages related to a topic. A browser extension enables the user to semi-automatically collage and reuse this content for personal informal learning. This content is then fed into a mobile learning application as digital flash cards. The usability of the complete workflow is of paramount significance. Therefore, this system seeks to optimize the workflow on several places.

The rest of the paper is organized as follows. First, we explain a workflow for ubiquitous micro-learning (Sec. II). The implementation of software components that comprise this workflow is described in Sec. III. Next we shortly report on a small evaluation (Sec. IV) and related work (Sec. V). In the final section we conclude our paper and give an outlook for future work.

II. Ubiquitous Micro-Learning

To be able to realize the application, it is necessary to understand the micro-learning process. Micro learning is a special learning method in the field of technology enhanced learning. There exist many different concepts and definitions for micro-learning, depending on the context. Hug [3] refers “to the communication technologies being used in the teaching and learning process”. Micro-learning means learning small units or simple issues in a measurable time or with relatively short effort. Instead of learning a complete topic only some small units like vocabulary or terms and definitions are learned. The learning process differs from the normal learning, because the content is learned in a small measurable time frame. The learning type is repetitive or learning by example or goal or problem oriented.

New technologies like Web 2.0, mobile devices and wireless web technologies are used to ease the learning and to simplify the learning process for the user. Therefore, micro-learning can be redefined as a ubiquitous learning activity on small pieces of knowledge based on web resources. Micro-learning is also about getting and organizing personally relevant information from many sources.

Figure 1 depicts a workflow that we follow. Each micro-learning activity is initiated within some concrete context, e.g. searching for a definition of some term. Users nowadays usually research on the Internet—the largest and most convenient collection of information.

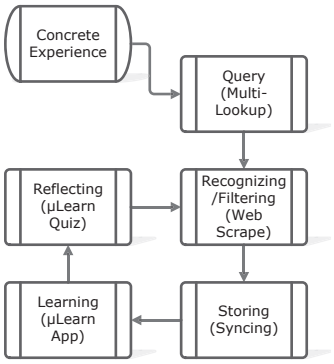


Fig. 1. Micro-learning workflow

In many cases, user's information needs cannot be immediately answered by a single web page or search results page, because these needs are complex and highly-personalized. This typically occurs when users query for certain topic in domains like learning, travel, health, where users need to consult with multiple heterogeneous web sources. For example, searching for meaning of some phrase in German for non-native speaker involves looking up in multiple translation, dictionary, thesaurus and example web sources. Donato et al. [4] showed that these "research missions" account for about 25% of the search volume on Yahoo! Search.

Each lookup, i.e. research mission, causes time consumption and mental efforts. Therefore, it deserves special attention and treatment. Federated search has been proposed in the research literature as a remedy. However, most of the proposed tools tend to be efficient only in certain closed domains such as flight ticket or hotel searching. We propose a browser extension called Multi-Lookup where the user can specify the sources for web search. The user can then join the individual sources (i.e. search engines) into groups which try provide comprehensive results for the intended search interest. This tool integrates within the browser search functionality. The user need to specify which search group, he/she wants to use for the next query. The result pages are opened in reusable tabs in the browser, i.e. subsequent queries use the already-opened browser tabs. The following is an example of a real user groupings of multiple search sources used in the daily life of a non-native English, non-native German speaker.

- Search engine group: English to English / Macedonian / German
  - thefreedictionary.com
  - rechnik.on.net.mk
  - translate.google.com
  - en.wiktionary.org
  - dictionary.reference.com
  - dict.cc
- Search engine group: German to German / English / Macedonian
  - duden.de
  - makedonisch.info
  - ...

With average size of five search engines per group, the amount of time saved in searching the same query in each engined individually is five-fold.

After identifying several result pages as potential answers to the search query, the user tries to identify a comprehensible answer. Our web content scraping tools enables users to collect relevant pieces of information from these result pages. It features visual highlighting of web page components (paragraphs, divs, articles, flash objects, images, headers, links, etc.) Parts of different web pages are selected by the user that he/she thinks they are most relevant in the current context. These parts (clips) are appended to the learning object (note). This tool reduces the time needed to collect chunks of web pages since it reduces selection operation to a single click on the highlighted element. Moreover, the collected clips are shown in the browser sidebar, thus preventing the user of loosing focus when switching between windows and tabs. The scraped content is then filtered, cleaned and automatically synchronized with the micro-learning mobile.

Our previous work [2] describes the conceptual workflow and web scraping in more detail. This paper focuses more on the implementation details of the mobile app.

III. Implementation

A. Cross-Platform Development

Developing an application for all mobile operating systems is very expensive and time consuming, because it is necessary to develop a new application for every system with the platform specific development kit. These problems are solved by Cross-Platform Developer Tools (CPT). They allow developing applications for different mobile systems with the same tool and with the same code base. The different CPTs use various methods to compile code for different mobile operating systems. These techniques result in three various kind of applications: native applications, web applications and hybrid applications. Currently, there are about 100 different Cross-Platform Developer Tools for mobile devices. They use different programming languages, different technologies and differ in many other important aspects like publishing possibilities and synchronization techniques [5].

RhoMobile was chosen to develop the µLearn application. Rhodes is a framework for developing applications optimized for mobile devices like apples iPhone or Android smartphones. The Rhodes framework is based on Ruby and works with the model view controller architecture. With this framework it is possible to build native device applications which can use device capabilities like compass, GPS, filesystem and camera. Synchronization and push notifications are supported with RhoConnect.

A Rhodes application is a web application that is wrapped by native code. So the web application runs locally on the device. A slimmed-down web server is running locally on the device. The different views of the application are specialized in template files, which contain HTML, CSS and JavaScript code. These template files are executed by the WebView control. Referred to the MVC pattern the model contains the data

structure that represents the state of the application. In this case, the model.rb file contains the model definitions. The controller acts between the model and the view. According to the user input, the controller changes the model data. In Rhodes the controller is a set of Ruby scripts. Many native device capabilities can be accessed with the controller [6].

RhoConnect is a Rho product for synchronizing all the data between an application and a backend server. Once data is changed on the server, the changed data is downloaded to the device. The same process is executed, when data is changed on the device. It is a two-way synchronization. The applications built with RhoStudio have a built-in client that is able to connect to a RhoConnect server. The RhoConnect server is kind of a bridge between the mobile application and the backend cloud data server. It can be installed on the cloud database server or a separate server. To connect the RhoConnect server with the backend server it is necessary to write "source adapters" which provide the methods login, logout, query, create, update and delete to call the backend server.

The µLearn application architecture is based on the Model-View-Controller (MVC) design pattern. The user interface is specified in the view with HTML, CSS and JavaScript. All data is saved in a local database. The RhoSync client and the controller can read and write data in the database. The view can read in the database only. The data from the local database is synchronized with the database on the µLearn server. The RhoConnect client does not access the cloud database server directly, instead it is accessed via the RhoConnect server. Therefore, the login, logout, query, create, update and delete operations are implemented in a source adapter on the RhoConnect server. The RhoConnect server sends also a push notification to the device when new data is available. The application logic is part of the controller. It connects the view and the database. The system architecture is shown in Figure 2.

The content is structured in notebooks and notes. Notebooks contain many notes and notes contain clips, which can be text, images or HTML excerpts. Notebooks and notes can be tagged. In this view the user can browse, view, create, delete, tag and edit notebooks.

B. User Interface

The application is divided into two different modes - browse and learn modes. A tabbar with a tab for each mode is created. So the user is able to switch between the two modes and the start screen anytime. The first mode is the "Browse" view. This mode provides functions for creating, editing and gathering information. The user can sort the notebooks by different criteria (s. Figure 3b). It is also possible to search the complete content for keywords. This increases the usability and makes finding notebooks or objects easy even if the user has a lot of learning objects. It is possible to edit every notebook, note and clip.

The second mode is the "Learn" mode. In this mode the user can learn the gathered information by selecting the notebook he/she wants to learn (s. Figure 3c). All clips which are marked as question will be displayed. Afterwards, he/she can click the button "show content". The content will be displayed and can check whether he/she knew all content (s. Figure 3d). If the user knew all content he/she rates the retention of the note on a 4 stars scale and the next term is displayed.

The learn view assists the user with learning his content. On the start page of the learn view all notebooks, which are not empty are shown. A pie chart is shown on the left side of every notebook. It shows the number of good rated (3 or 4 stars) in green, number of bad rated (1 or 2 stars) in red and the number of new notes in black. This should give the user an overview of the learn status of the different notebooks. It also should help him to select a notebook for learning, because he can directly see which notebooks he knows best.

A note can be a question or a fact. Both kind of notes are treated different in the learn view. Both views display the name of the note in the "topbar". In the topbar is also a back button and a skip this question button. Question notes can have special question clips, which contain a question. Only these clips are shown first. The user reads the question and thinks about all the stuff he knows about this topic. Afterwards, he can click on the "show content" button, which is shown under the questions. Clicking on tab will show all the content of this note. If the selected note is a fact all content is displayed directly.

In the "learn view" there is also a fixed bar on the bottom, where the user can rate the note. This bar is visible all the time, because the user should be able to rate the note anytime. When the user is finished with learning a note, he can rate the difficulty of this note and the next note is shown. The user can also mark a note as learned, when he does not want to learn this question anymore. If the user thinks that the notes does not contain enough information, he can mark this question as "enhanced". Everytime the user rates a note, a new entry is added to the statistics database. This entry contains information about time and rating. The rating is also updated in the notes database.

There are two different methods for selecting a note of the chosen notebook. If the notebook has less than fifteen notes, one note of the notebook is selected randomly. Notes with a low rating have a higher chance to be chosen than a note with a higher rating. A note rated with one star gets a weight of four and a note with four stars gets a weight of one. If the notebook has more than fifteen notes, the space repetition "Super Memo 2" algorithm (SM2) is used [8]. Therefore, a new datamodel called "CardDeck" is created. CardDeck contains up to ten notes for every notebook. This is necessary, because the spaced repetition algorithm works with card decks. The last rating, interval, eFactor and learncount are saved for every note. This information is necessary for the spaced repetition algorithm. The eFactor can vary between 1.3 and 2.5. The less the value the more difficult the note. Every new note gets an initial eFactor of 2.5. The interval indicates the time in days when this note should be repeated. The interval is calculated with the help of the eFactor and the number repetitions ("learncount"). After a note is rated by the user, the "learncount" is increased

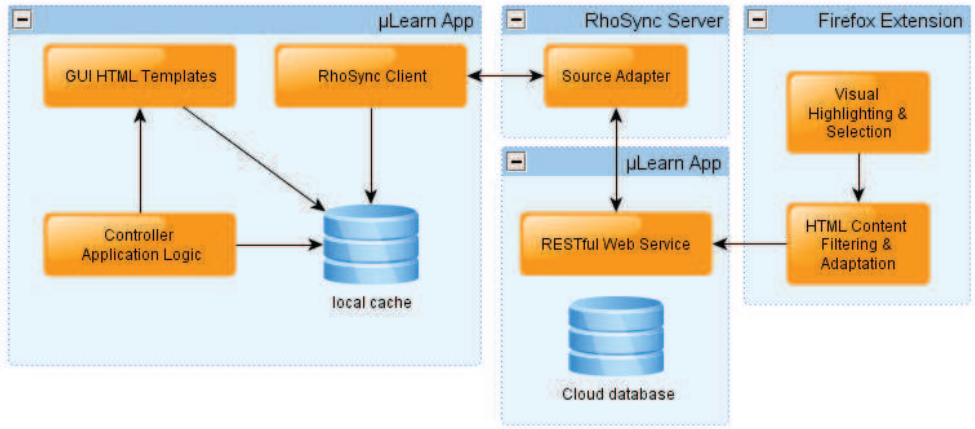
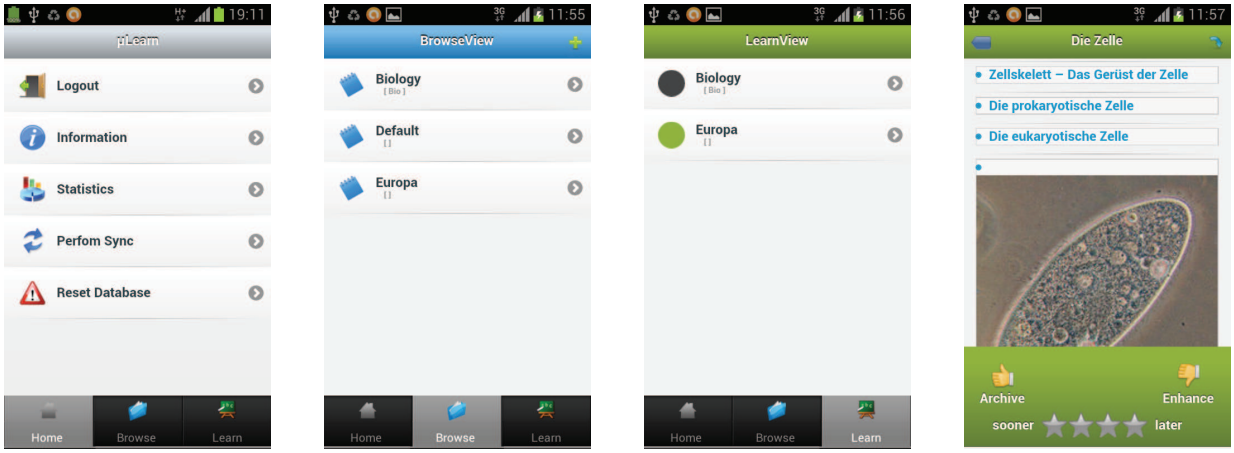


Fig. 2. System Architecture



(a) Home screen (b) Notebooks browsing (c) Notes browsing (d) Note learning

Fig. 3. Simulation results.

by one, the new eFactor and interval is calculated.

C. Synchronization

The synchronization is implemented with the RhoConnect server and client. The client is part of the RhoMobile application and the server is a special framework which runs on a Ruby server. The server enables the data exchange between the application and the database server. The data is exchanged in the JavaScript Object Notation (JSON). The RhoConnect server can not access the database directly. So the database server needs to supply a web service. The web service is implemented with a PHP 5 micro framework called "slimframework". "GET", "POST", "PUT" and "DELETE" requests are used to request, add, update and delete data from the database. In addition a login and logout function is offered. If the login data is correct a sessionId is generated and sent to the device. All other functions can be executed with a valid sessionId only.

For every database table, which should be synchronized, a new "source adapter" needs to be implemented. A "source adapter" specifies the query, create, update and delete functions and connects so the "RhoConnect server" with the webservice. The RhoConnect application uses the Ruby rest client to send the different requests to the webservice.

The RhoConnect server should be able to inform any connected device about new or changed data. Therefore, the server uses the device specific push notifications service, e.g. iOS Push Notifications or Google Cloud Messaging for Android. For every mobile operating system information about the push notifications service needs to be set up in the settings.yml of the RhoConnect server. The RhoConnect push server is a separate process which sends message between the RhoConnect server and the registered devices. The application needs to be able to receive and to handle the push notifications.

IV. Evaluation

The application was evaluated to get to know whether the application is easy to understand and intuitive to use. To be able to analyze the user evaluation in an appropriate way, a structured questionnaire was created. The questionnaire consists of questions regarding the design, the usability of the application and some information about the user. The application and the questionnaire were handed to six test users. Both experienced and non experienced users were chosen.

After one week the filled out questionnaire was collected and evaluated. The focus of the user evaluation is the usability of the application and not the learning effect. The test user should focus on using the application. So some example content and instruction manual were created. It included a video, where the user could see how the content is selected from the Web.

The test users were intermediate to expert smartphone users and most users had an age between 16 and 25. The age group of 26-35 and 46+ was covered, too. Every user had a different device, so the application was tested on six different devices with also different OS versions. Most of the users used the application 10 to 20 times in the given time frame. All users understand the application and even the meaning of the most icons was clear. The appearance of the "skip" button caused some trouble. All participants explained the differences between browse and learn view correct. The results show that the design and the structure of the app are intuitive, the test users had no problems with using the application. Also the user understood the meaning of the different views. The used icons are intuitive to use, except the "cancel" and "skip" button. These button were redesigned to a button which contains text.

V. Related Work

Our micro-learning app is a combination of quiz (flash card) and note taking functionalities. There are many different learning applications in the iOS and Android market, e.g. apps for learning grammar and vocabulary of different languages or mathematics content. Some other applications test the general knowledge of the user with a quiz. All these applications differ from the μLearn application, since all use a fixed knowledge database. The content and quizzes are static and set by the developers of the application. Two examples of learning apps with static content are "English in Pictures Food Free" [9] and "World Countries: Quiz and Learn"[10]. The μLearn application instead allows the user to create his own content and quizzes. That enables the user to learn his personally-relevant content. This results in an individual way of learning.

Micro-learning with mobile devices has already been studied. The MicroMandarin [11] project examined how flash cards can support mobile micro-learning of second-language phrases presented in context, investigating what material should be studied where. Edge et al. [12] studied the effectiveness of text and audio modalities in micro-learning context. They showed that micro-learning is important addition to a learner's repertoire. Applications like Evernote [13] offer similar functionalities like our micro-learning system, but they provide general features like note taking. Learning features like questionnaires and spaced repetition are not provided.

All these approaches lack the consideration of the complete micro-learning workflow beyond the mobile devices. Mobile devices are suitable for learning and practicing at non-formal learning location locations. However, mobile devices are not suitable for production of learning content. With our browser extensions for searching and content scraping we complement the mobile client apps with fast and flexible creation of learning content.

VI. Conclusions

The μLearn application gives users the possibility to learn content that is selected from the Web with a Firefox browser extension. It also offers the possibility to edit and to create content on mobile devices. To be able to implement this application for Android and iOS the cross platform developer tool RhoMobile was used. Synchronization was realized with RhoConnect tool included in RhoMobile framework.

Future version of μLearn application could include more different learning strategies. So users can select their favorite learning strategies. Strategies are used for the selection of notes and they regulate the repetition of notes. Another nice-to-have feature is learning notifications. The application informs the user about the notes he should learn today. This can increase the learning effect, because the user repeats all notes after a special time period, depending on the number of repetitions and the rating. So the user has more possibilities to see a more detailed learning curve. Another useful feature would be the sharing of learning statistics or private learning notes on Facebook and other online communities.

References

[1] G. Gassler, T. Hug, and C. Glahn, "Integrated micro learning–an outline of the basic method and first results," in *Proceedings of the Interactive Computer Aided Learning (ICL 2004)*, 2004.

[2] D. Kovachev, Y. Cao, R. Klamma, and M. Jarke, "Learn-as-you-go: New ways of cloud-based micro-learning for the mobile web," in *Lecture Notes in Computer Science Advances in Web-based Learning - ICWL 2011*, H. Leung, E. Popescu, Y. Cao, R. W. H. Lau, and W. Nejdl, Eds. Springer, 2011, vol. 7048, pp. 51–61.

[3] T. Hug, "Micro Learning and Narration - Exploring Possibilities of Utilization of Narrations and Storytelling for the Designing of "micro units" and Didactical Micro-learning Arrangements," in *The Fourth Media in Transition Conference (MiT4)*, May 2005.

[4] D. Donato, F. Bonchi, T. Chi, and Y. S. Maarek, "Do You Want to Take Notes?: Identifying Research Missions in Yahoo! Search Pad," in *Proceedings of the 19th International Conference on World Wide Web (WWW 2010)*, 2010, pp. 321–330.

[5] M. Kapetanakis, "Cross-Platform Developer Tools 2012: Bridging the Worlds of Mobile Apps and the Web," www.crossplatformtools.com, 2012, [Online; accessed 23-May-2012].

[6] A. Nalwaya, *Rhomobile beginner's guide: Step-by-Step Instructions to Build an Enterprise Mobile Web Application from Scratch*. Birmingham and UK: Packt Pub., 2011.

[7] "Rhomobile Documentation," http://docs.rhomobile.com/, 2012, [Online; accessed 21-September-2012].

[8] "Super Memo," http://www.supermemo.com/, 2012, [Online; accessed 21-September-2012].

[9] "English in Pictures Food Free," https://play.google.com/store/apps/details?id=com.picture.learn.english.trial, 2012, [Online; accessed 30-May-2012].

[10] "World Countries:Quiz and Learn," https://play.google.com/store/apps/details?id=com.DiSPi.QaL_Countries, 2012, [Online; accessed 30-May-2012].

[11] D. Edge, E. Searle, K. Chiu, J. Zhao, and J. A. Landay, "MicroMandarin: Mobile Language Learning in Context," in *Proceedings of ACM CHI Conference on Human Factors in Computing Systems (CHI'11)*. ACM, 2011, pp. 3169–3178.

[12] D. Edge, S. Fitchett, M. Whitney, and J. Landay, "MemReflex: Adaptive Flashcards for Mobile Microlearning," in *Proceedings of the 14th ACM Conference on Human-Computer-Interaction with Mobile Devices and Services (MobileHCI 2012)*, 2012.

[13] "Evernote," http://evernote.com/intl/de/, 2012, [Online; accessed 30-May-2012].