ReqVision: Digitising Your Analog Notes into Readable and Editable Data

Natalia Mannov
University of Hamburg
Department of Informatics
Germany, Hamburg
mannov@informatik.uni-hamburg.de

Clara Marie Lüders
University of Hamburg
Department of Informatics
Germany, Hamburg
lueders@informatik.uni-hamburg.de

Alexey Kaznin
Northern (Arctic) Federal University
Department of Applied Informatics
Russia, Arkhangelsk
a.kaznin@narfu.ru

***Abstract*—The paper describes the problem of collecting, merging, and sharing information from workshop and presentation notes in order to achieve better cooperation between stakeholders, thus improving collaboration during product development. We provide an overview of apps for Android and iOS devices that can facilitate this problem. We tested these apps on their recognition of handwritten and printed text and their integration into existing services, but none of them met all desired requirements. We propose an approach that includes a self-adaptive part to improve handwritten text recognition as well as a context-sensitive part to facilitate the integration into existing systems in the context of the *Smart RE* framework [1].**

I. Introduction

Workshops and presentations are important to elicitate, clarify, and negotiate requirements. During such events it is hard to take notes, listen, and participate actively at the same time. Requirements engineers often dislike any digital support via laptop, smart board or tablet, especially in workshops. Moreover, using smart devices within a workshop or presentation can be distracting for all participants caused by negative effects of multitasking and interruptions which influence the interaction quality [2]–[5]. Furthermore, writing is better suited for taking notes than typing [6] and many people prefer drawing or writing. Even though typing is faster than writing and changes can be committed easily, typing restricts the freedom of layout and design, thus decreasing creativity.

To stay creative and efficient, requirements engineers apply different methods to produce usable output like text descriptions, diagrams, or sketches. A wide variety of means to capture the output exists, e.g. notepads, presentation cards, whiteboards, pinboards or flip charts. These types of media are perfect to quickly create and dismiss new ideas. At the end of these events, requirements engineers can collect the reliable results and use them for further work. Requirement engineers usually collect these results by using their smartphone to take a picture and either save it on their device or send it to themselves via email. These steps depend on the company's security policies and the availability of technical support. Different physical sources, output formats, communication systems, issue tracking and project management systems across the different stakeholders can cause chaos and lead to the loss of time and information.

To lessen the chaos and prevent loss of information, digitising notes is a viable option. Unfortunately, this process is not currently seamlessly integrated in the workflow of requirements or software engineers. Our semi-structured interviews across eight experts, who are involved in requirements engineering activities, reveal the necessity to bridge the gap between the many available different technical and software solutions.

Many tools and approaches to analyse, compare, and process text files and predict, reveal, and create relationships based on the textual content exist. All of them have in common that they are only able to process editable or at least readable text. This is a difficult problem to solve, due to the poor results of processing handwriting into the required text format.

Many smartphone apps are available on *Google Play Store* and *Apple App Store* which can be used to digitise printed and handwritten data, not necessarily in a Requirements Engineering or Software Engineering context. We evaluated ten different smartphone apps, particularly the user-friendliness, processing time, available functionality for picture pre-processing, printed text recognition, handwritten text recognition, and integration into the workflow and existing systems.

We depict our vision of the application based on two user scenarios in Section II. In Section III we present the results of our app evaluation. Section IV discusses our results and in Section V we present a solution based on our vision of ReqVision smartphone app and the *Smart RE framework* [1].

II. ReqVision App - The Vision

Figure Fig. 1 depicts our vision by describing two user scenarios: a workshop situation and a presentation situation. The user story workshop situation can be also applied for a meeting situation.

A. Workshop or Meeting Situation (see Fig. 1, first row)

Alice is a requirements engineer and today her job is to identify the crucial requirements for a big project of a new customer. The customer wants to obtain a communication system for their ambulance care service. Alice invited customer representatives, the product owner, a few developers, three ambulance care assistants and two hospital nurses to the workshop. The workshop started with a discussion on

the end product using handouts. Then, participants drafted an improved workflow of the involved stakeholders based on their practical experience. During the workshop, they outlined on the whiteboard the typical workflow of ambulance care assistants, developed user stories, and created story cards. They collected several ideas for specific features of the end product on the pinboard and wrote down relevant requirements (see Fig. 1 A). Ambulance care assistants and nurses mentioned several requirements dealing with the coordination required to transfer patients. There is the need to simplify the process by clarifying the responsibilities and the required actions. Additionally, the nurses need timely support from the ambulance care assistants to document transfers and schedule next actions according to the diagnosis.

After the six hour workshop, Alice and Ben, a software developer in the same project, collected and sorted all produced results. Alice pulls out her smartphone, opens the ReqVision app, takes a picture of the handwritten requirements on the flip chart (see Fig. 1 B). The ReqVision app provides the recognised text parts for editing. Alice improves and corrects the content, formats the text, and makes notes. The system suggests possible labels and references for related issues, user stories, existing requirements and documents. Alice assigns labels, appropriate references, user access rights for other project members and saves the content to the project management system (see Fig. 1 C). Alice and Ben discuss the end product requirements, described in the printed version, and

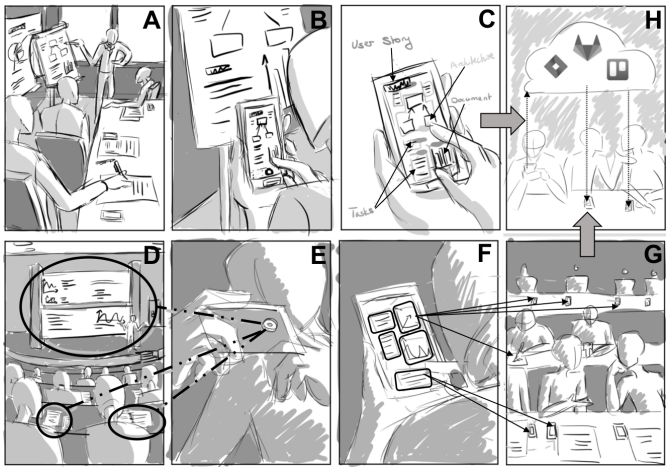

Fig. 1: User scenario "*Workshop*" and "*Presentation*". *A.)* Participants are sketching and discussing the workflow of the end user. *B.)* Alice documents the workshop output. *C.)* Alice orders and labels data in the captured picture, then saves it in JIRA. *D.)* Participants read presentation handouts and make notes for the presentation. *E.)* Alice takes a photo of current presentation slide and her notes. *F.)* Alice comments selected parts of captured picture and saves it in JIRA. *G.)* Alice shares her noted thoughts attached to the captured picture with her colleagues for further discussion, then saves it in JIRA. *H.)* Developer team discusses new insights at their next meeting.

the requirements the ambulance care assistants and hospital staff revealed during the workshop. They are ambiguous and inconsistent. So Alice marks the corresponding text passages and attaches the page on the whiteboard, takes a picture with the ReqVision app, selects the text passages and two related statements on the whiteboard. She adds it to the appropriate issue as a picture and shares the extracted text file with Ben and the product owner for further discussion.

She has to hurry to catch a flight to Florence. On account of this she takes a picture of each relevant piece of produced workshop results on the whiteboard, pinboard, and flip chart.

It turns out that the flight is delayed. While waiting, Alice extracts texts, graphs, and diagrams by selecting and editing pictures taken during the workshop and then saves the results in the project management system. Using the ReqVision app, she can edit, label, comment, link, and add pictures (or parts of them) to the related project documents and issues. Despite her absence, Alice's colleagues can keep working on these new insights. For example, they can create useful and meaningful use cases and user stories for further development of the end product (see Fig. 1 H).

B. Presentation Situation (see Fig. 1, second row)

Once she arrives in Florence, Alice attends the annual board meeting of her company. This year, the focus is on insights into new technologies and methods for communication support. A lot of new ideas and approaches are presented. She gains new insights and participates in many interesting discussions. The next presentation contains some particularly relevant information for her current project. She reads the abstract, marks noteworthy text passages, and notes some project-related terms (see Fig. 1 D). During the presentation, she opens the ReqVision app and takes a picture of a slide with an interesting graph. The slide shows some new handy devices, which could address the patient transfer problem experienced by nurses and ambulance care assistants (see Fig. 1 E). She takes a picture of the handouts and of her notes, selects related areas of text, and lets the ReqVision app digitise them separately (see Fig. 1 F). Afterwards, she improves and adjusts the recommended text, comments on it, and shares it with Ben. By opening a new chat, she starts a discussion about the graph and its related information (see Fig. 1 G).

Finally, she opens the picture of her notes on the presentation handout and the recognised text, she comments, assigns labels, links everything with the scanned data graph and saves it to her current project in the project management system. Now her colleagues are able to discuss the new input from the presentation in their next meeting tomorrow before she will return next week (see Fig. 1 H).

III. Review of Selected Available Smartphone Applications

We looked at available smartphone apps in the Google Play and Apple App Store with appropriate features for the user scenarios described above. We selected and tested ten smartphone apps (see TABLE I). This section gives a short

overview of how we searched for apps, how we tested them and the results we found.

A. Filtering Criteria

We searched for smartphone apps available for Android and iOS devices that provide the functionality to scan text from a picture. We used the search-string "text recognition" and filtered the results for free apps on the Google Play Store [7] and TheAppStore.org [8]. Both resources allow to search for apps directly from the browser. Additionally, the results can be filtered by rating, update date, and price.

The Google Play Store listed 250 apps for our search string. From the app description, 49 of these apps allow to recognise printed text. Additionally, four of them allow to recognise handwritten text and two other objects. The TheAppStore.org [8] provided 100 apps using the same search string in total, 59 of them are free. From the app description, 26 apps provide text (22 of 26) and object (four of 26) recognition. Additionally, we searched for the search string "handwritten ocr" in the Apple App Store on iPhone 6. The search provided three free apps: *Scan & Scribe - Phone Docs OCR*, *OCR Handwritten*, and *Pen to Print - Handwriting OCR*.

B. Evaluation of Smartphone Applications

We tested four apps available for Android and iOS, four apps available only for Android devices, and two apps available only for iOS devices, see TABLE I. Apps were tested on pictures taken with the smartphone camera of handwritten and printed text samples. Furthermore, we tested the apps on imported pictures of a printed text in Latin with different quality: 1) taken from a paper, 2) taken from the display screen with persistent noise and light reflections, 3) taken from a paper with blurred character boundaries [9]. Two native users of the operating systems iOS and Android tested the apps on their corresponding devices, an iPhone 6S and a Samsung S5 Neo, and collected the results.

All apps can share the results either by copying and sending the text via email or by a share button via a selected app on the smartphone or by saving it as a document, e.g. PDF or as an image on the smartphone. *Office Lens* additionally provides the feature to open the recognised text for further editing in a Word or PowerPoint document. But no app offered to save the produced files directly in a project management system. All of them are able to recognise printed text. They all provide the result within 1-3 minutes. The accuracy of the results varies strongly from app to app and depends on the quality of the input picture. The best results for all three printed text samples were provided by *Pen to Print - Handwriting OCR*. The recognised text included only few character errors. Unfortunately, the results view and its editability are subpar. The app provides recognised text editable row by row, but if the row has more than 40 characters it is very frustrating to edit the text. *CamScanner* is the fastest app to recognise the printed text, but regrettably the accuracy of the results are correspondingly bad; the origin text is unrecognisable. *CamScanner* suggests crop boundaries for the text part selection automatically. Several

TABLE I: Overview of tested smartphone apps

App name	Andr.	iOS	Printed text	Hand-written	Rating Andr./iOS
CamScanner	+	+	+	-	4.6 / 4.7
Office Lens	+	+	+	-	4.7 / 4.7
Scan & Scribe- Phone Docs OCR	+	+	+	+	3.0 / n.s
Scanner&Übersetzer Foto to Text	+	+	+	+	3.5 / 4.3
Google Goggles	+	-	+	+	3.9 / —
Text Recognition And Translate	+	-	+	+	4.6 / —
Smart Lens	+	-	+	+	4.5 / —
Text Scanner [OCR]	+	-	+	+	4.5 / —
OCR Handwritten	-	+	+	+	— / n.s
Pen to Print- Handwriting OCR	-	+	+	+	— / n.s

tested apps allow varying different image properties and settings to get better end results. Only *Scan & Scribe* describes several restrictions and weaknesses of the used recognition algorithm. Usually there are the following: difference between typed text, disjoint and connected handwriting; image text direction, interference factors like any lined, grid-lined or multicolour background, other lines or drawn objects beside text characters, blurred or deformed images, light and shadow conditions, light reflections and permanent noise and input language. A few of the tested apps allow the user to define the target language for recognition, e.g. *Scanner&Übersetzer*, *Scan&Translate*.

Most of the tested apps are also applicable on handwritten text samples. Sadly, the results are poor. *Office Lens* simply captures an image of the handwritten text sample and *CamScanner* shows an error message after a specific limit of processing time. *OCR Handwritten* (iOS) and *Smart Lens* (Android) provided the best results, if the handwriting is clear.

Summarising, the current smartphone applications offer some necessary functionalities to solve the problem described in the introduction, but essential features are still missing. Handwriting recognition is only very limited supported, there are no self-adapting components that facilitate handwritten text recognition or any components that recognize the different components of a document (e.g. text, pictures, tables). Additionally, the integration into the workflow and the ability to share information is still lacking, the process in the evaluated apps is cumbersome, a self-adaptive and context-sensitive method that learns to recommend where and how to save and share files is needed to ease this process. Also, an offline mode was only offered in less than half of the tested apps.

IV. Discussion

We want to develop an app for Android and iOS based on the idea of Smart RE framework [1], but there are several challenges we face to create an app which solves the problem described in the introduction.

Established OCR algorithms work well on pictures with high picture quality, clear character boundaries, and in the absence of any disturbing factors e.g. permanent noise, light reflections, shadows, and blurring. In real working conditions

this input quality is impossible to achieve. This impacts the results of the text recognition, but image pre-processing can improve the results significantly. Pre-processing needs large computing resources and modern mobile devices have high-performance, due to their many processing cores, additionally, parallel data computing improves the pre-processing time. The big challenge is to rebuild existing pre-processing algorithms into parallel algorithms.

To extract readable and editable text from the text blocks of an image, we need a versatile and reliable handwritten text recognition. A high quality printed text recognition is already provided by several services. But the current status of the handwritten recognition in apps leaves much to be desired. It barely worked and only if the handwriting is clear and recognisable, which is usually not the case for handwritten notes. Therefore, we want to compare the approaches in [10] and [11]. Additionally, a self-adaptive component can help solve this problem. Another way to improve the output of the text recognition is the use of the target language. This information can improve predictions since it limits the possible results. By specifying the target language we deal with a specific and smaller symbol set and vocabulary.

The app should offer the choice to process the inputs online and offline. Due to security standards of the companies – especially within Europe, the app is not allowed to save the images in another location to process them. This problem might result in a longer processing time since all inputs need to be processed on the smartphone. In some cases the reduced usability due to different processing powers of smartphones might outweigh the gained security. Therefore, the app should also offer the ability to process the inputs online. This way public data can be processed faster online and private data can be processed offline. Another problem arises from this functionality, if we use services, like Google Mobile Vision [12] or Microsoft Computer Vision [11], then these processes will not occur on the smartphone. We can only overcome this obstacle by integrating and developing the functionality of these services by ourselves.

The last challenge is the integration of the app into the workflow of the users. Ideally, we want the app to share the output via copied text, as a word, text or PDF file. The users should also be able to store the output file to their respective project management system, e.g. JIRA or GitLab, without any intermediate steps. Also, the app should learn what items inside the project management system are relevant to the outputs produced by the app and suggest these items to the user.

V. Future Work

We want to develop an app for Android and iOS based on the idea of Smart RE framework [1]. The Smart RE Device App, here called ReqVision app is able to load input data, either by taking a new picture or selecting an image from the device's storage. The next step would be the document layout analysis, the different parts of the image should be recognised into blocks, i.e. texts, diagrams, pictures, and tables. After the pre-processing - applying several operations to clear up the image and raise the image quality: converting the image into greyscale, filtration, correction of different brightness, binarization and rotation. The user can then group and label the blocks or adjust the recognition to his liking. If the users choose to not perform any text recognition, they can save the created output into a file, which can be shared or uploaded to respective project management tool. If the user chooses to use text recognition, the app will provide suggestions for the text blocks which can be edited and improved. The app should be able to learn from these edits to deliver better predictions the next time it is used. Afterwards, the user can choose how to save and share the generated file. Additionally, if the app is integrated into the project management system, the system suggests fitting items from the project management system to attach the file to.

Acknowledgment

The authors would like to thank Isabella Tran for providing the figures and her support in developing the concept for the app. This work is partly funded by the H2020 research project OpenReq (grant agreement No. 732463).

References

[1] N. Mannov and W. Maalej, "Smart re: using smart devices to support face-to-face meetings," in *2015 IEEE Workshop on Just-In-Time Requirements Engineering (JITRE)*, pp. 9–12, Aug 2015.

[2] H. Hembrooke and G. Gay, "The laptop and the lecture: The effects of multitasking in learning environments," *Journal of Computing in Higher Education*, vol. 15, pp. 46–64, Sep 2003.

[3] L. M. Carrier, L. D. Rosen, N. A. Cheever, and A. F. Lim, "Causes, effects, and practicalities of everyday multitasking," *Developmental Review*, vol. 35, pp. 64 – 78, 2015. Special Issue: Living in the Net Generation: Multitasking, Learning, and Development.

[4] E. R. Sykes, "Interruptions in the workplace: A case study to reduce their effects," *International Journal of Information Management*, vol. 31, no. 4, pp. 385 – 394, 2011.

[5] M. M. V. Abeele, M. L. Antheunis, and A. P. Schouten, "The effect of mobile messaging during a conversation on impression formation and interaction quality," *Computers in Human Behavior*, vol. 62, pp. 562 – 569, 2016.

[6] P. A. Mueller and D. M. Oppenheimer, "The pen is mightier than the keyboard: Advantages of longhand over laptop note taking," *Psychological Science*, vol. 25, no. 6, pp. 1159–1168, 2014. PMID: 24760141.

[7] Google, "Android Apps on Google Play, month = , year = , note="[online]. available: https://play.google.com/store/apps. [accessed: Jun-2018]", url = https://play.google.com/store/apps."

[8] Search and filter iTunes Store on web browser |The App Store, "TheAppStore.org, month = , year = , note="[online]. available: https://theappstore.org. [accessed: Jun-2018]", url = https://theappstore.org."

[9] A. A. Kaznin, O. P. Sushko, and A. V. Babkin, "Developing the algorithm allowing business-dedicated mobile applications to read texts," in *2017 International Conference "Quality Management,Transport and Information Security, Information Technologies" (IT QM IS)*, pp. 207–214, Sept 2017.

[10] T. Bluche, J. Louradour, and R. Messina, "Scan, attend and read: End-to-end handwritten paragraph recognition with mdlstm attention," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, pp. 1050–1055, Nov 2017.

[11] Service Level Agreements Summary |Microsoft Azure, "Computer Vision [RU] |Microsoft Azure." [Online]. Available: https://azure.microsoft.com/ru-ru/services/cognitive-services/computer-vision/. [Accessed: Jun-2018].

[12] Google, "Mobile Vision |Google Developers." [Online]. Available: https://developers.google.com/vision/. [Accessed: Jun-2018].