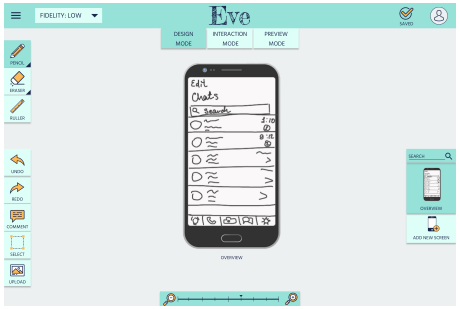

Figure 1: Eve: Low Fidelity

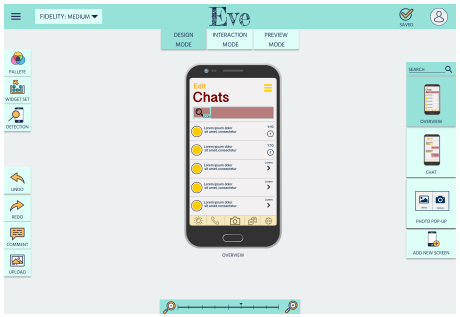

Figure 2: Eve: Medium Fidelity

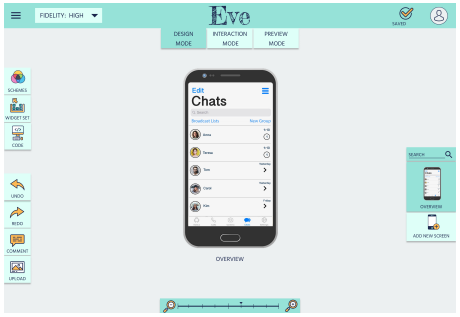

Figure 3: Eve: High Fidelity

Eve: A Sketch-based Software Prototyping Workbench

Sarah Suleri
Fraunhofer FIT UCC
Sankt Augustin, NRW, Germany
suleri@fit.fraunhofer.de

Vinoth Pandian Sermuga Pandian
Fraunhofer FIT UCC
Sankt Augustin, NRW, Germany
pandian@fit.fraunhofer.de

Svetlana Shishkovets
Fraunhofer FIT UCC
Sankt Augustin, NRW, Germany
shishkovets@fit.fraunhofer.de

Prof. Dr. Matthias Jarke
Fraunhofer FIT UCC
Sankt Augustin, NRW, Germany
matthias.jarke@fit.fraunhofer.de

ABSTRACT

Prototyping involves the evolution of an idea into various stages of design until it reaches a certain level of maturity. These design stages include low, medium and high fidelity prototypes. Workload analysis of prototyping using NASA-TLX showed an increase in workload specifically in frustration, temporal demand, effort, and decline in performance as the participants progressed from low to high fidelity. Upon reviewing numerous commercial and academic tools that directly or indirectly support software prototyping in one aspect or another, we identified a need for a comprehensive solution to support the entire software prototyping process. In this paper, we introduce Eve, a prototyping workbench that enables the users to sketch their concept as low fidelity prototype. It generates the consequent medium and high fidelity prototypes by means of UI element detection and code generation. We evaluated Eve using SUS with 15 UI/UX designers; the results depict good usability

CHI'19 Extended Abstracts, May 4–9, 2019, Glasgow, Scotland UK

ACM ISBN 978-1-4503-5971-9/19/05.
https://doi.org/10.1145/3290607.3312994

KEYWORDS

Interface design prototyping; User Interface design; User centered design; Software prototyping process; Rapid application development; Graphical User Interfaces

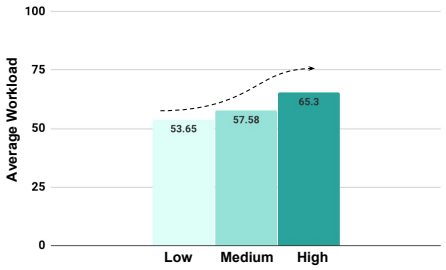

Figure 4: Average workload summary for Low, Medium and High fidelity prototyping

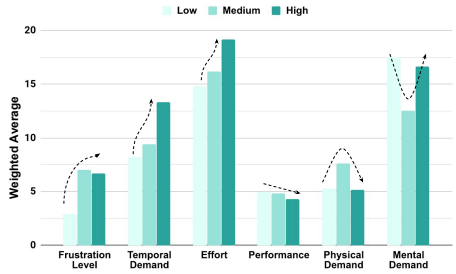

Figure 5: Subscale summary of Low, Medium and High fidelity prototyping

and high learnability (Usability score: 78.5). In future, we aim to study the impact of Eve on subjective workload experienced by users during software prototyping.

INTRODUCTION

Prototyping is a widely used method of designing and evaluating user interfaces. It consists of three fidelities: low (LoFi), medium (MeFi) and high (HiFi) [5]. Each fidelity refers to the maturity level of the prototype. The prototyping process entails taking a concept from a rough draft to eventually turning it into something substantial that can be evaluated by the target users. This process is iterative, and therefore, takes time to reach a level of maturity. Like any other creative process, UI/UX designers often have to design and redesign concepts until a consensus is reached. The design iterations of each fidelity and rework involved in transforming one fidelity to another can make this entire process strenuous.

In our research, we intend to investigate software prototyping from the workload perspective. Here, the term *workload* not only refers to the amount of work to be performed during the task, but also to the level of physical and cognitive burden experienced by the person [6]. Furthermore, we aim to support software prototyping technologically and compare the subjective workload experienced by users in the current (*as-is*) and automated (*to-be*) scenarios.

INITIAL WORKLOAD ANALYSIS

To further our investigation regarding the *as-is* scenario, we studied the subjective workload experienced by UI/UX designers during software prototyping. For this purpose, we conducted a workload analysis study with 18 participants using NASA Task Load Index (NASA-TLX) [7]. Participants were asked to prototype a given concept and report the experienced workload for each fidelity. As per Figure 4, we observed that the average workload experienced by the participants increased with each fidelity of prototyping. Subscale summary of workload analysis in Figure 5 shows an increase in frustration, temporal demand and effort, whereas, a decrease in performance as the participants progressed from low to high fidelity. Comparatively, MeFi is physically more and mentally less demanding than both LoFi and HiFi.

PROTOTYPING TOOLS REVIEW

We focused on reviewing existing tools on the basis of their support for LoFi, MeFi and HiFi prototyping. Upon reviewing literature and existing tools, we found 141 tools that support prototyping: 118 commercial tools, and 23 academic artifacts. Among these tools, there are 21 drawing tools that are not prototyping tools but afford designing LoFi prototypes. There are 19 MeFi prototyping tools which support LoFi prototyping to a limited extent by providing drawing features using bezier, pencil tool or basic shapes. Overall, 56 prototyping tools focus on MeFi as the starting point of the prototyping

process; using *Drag and Drop* for designing and *hotspots* or *hyperlinks* for creating interactions. We observed that most prototyping tools do not support designers to work on a LoFi prototype, but instead expect them to start prototyping from MeFi. To support HiFi prototyping, 28 tools generate executable code against MeFi design; whereas, 4 tools only provide code snippets. In summation, most tools look at prototyping as distinct steps, but in reality they are interconnected and therefore, require comprehensive tool support.

To further our investigation, we looked into existing tool support for semi-automating the transformation of LoFi to HiFi prototype. Among commercial tools, Airbnb's sketching interfaces tool [15] and Microsoft AI.Lab's Sketch2Code [14] support the conversion of LoFi design to code using machine learning. Airbnb's sketching interfaces tool [15] is the only tool that keeps sketching on paper as the main input for prototyping and detects drawn sketches via web cam to convert it finally to code. This tool is not available for review; the information is only published via demo video. Sketch2Code [14] takes a LoFi sketch image as input and converts it to HTML code. It does not support the transformation of interactions. Besides these two tools, pix2code [1] supports transformation of MeFi design to code using machine learning. As per the published information, robustness of the system is not evident. In academic tools: SketchiXML [4] and UsiSketch [11] use a visual grammar based stroke recognizer; Freeform [12], Java Sketch-it [3] and UISKEI [13] use pattern recognition to detect strokes. These academic artifacts use out-dated pattern recognizers and are currently not available for use.

Conclusively, despite the existence of numerous tools that directly or indirectly support software prototyping in one aspect or another, there is a lack of an all-in-one system that supports the entire software prototyping process comprehensively.

EVE: A SKETCH-BASED PROTOTYPING WORKBENCH

In this paper, we introduce Eve, a workbench that automates software prototyping. As per our preliminary semi-structured interviews with 18 UI/UX designers, 88% showed inclination towards starting the design process by sketching their ideas as LoFi prototype. Interviewees found sketching as a quick and dirty way to jot down ideas. The initial workload analysis also manifested that designers experienced the least workload for LoFi as compared to other fidelities. For these reasons, unlike Sketch2Code [14] and pix2code [1] where users have to upload an image to generate code, we decided to have sketching as the main input from users.

In Eve, the idea is to provide users with a *Canvas* to draw the application concept (Figure 1). In the background, the *UI Element Detector: MetaMorph* detects the sketched UI elements using Deep Neural Networks (Figure 8). With this information, the *UI Element Generator* generates the respective UI elements as MeFi prototype (Figure 2). Lastly, the *Code Generator* transforms the MeFi to HiFi by generating executable code (Figure 3).

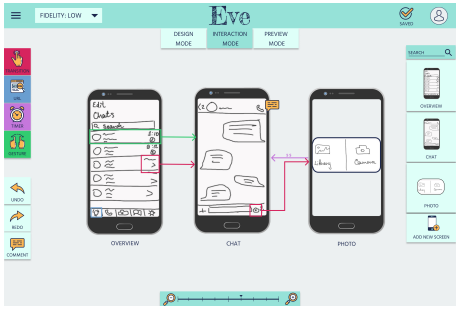

Figure 6: Eve: LoFi Interaction Mode

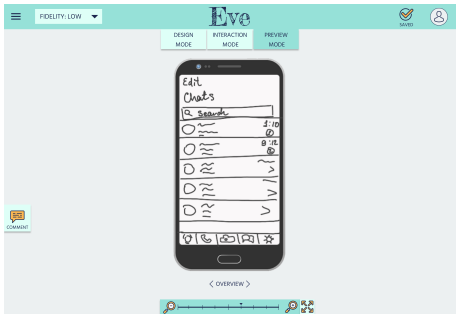

Figure 7: Eve: LoFi Preview Mode

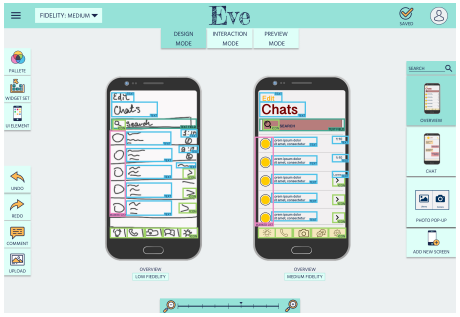

Figure 8: Eve: LoFi to MeFi Transformation

Users can navigate through low, medium and high fidelity and make desired changes at any point in time. Each fidelity entails three modes: design, interaction and preview. Each mode provides users with various features to support prototyping tasks for that particular fidelity. This feature list was generated as a result of reviewing literature, existing prototyping tools and preliminary interviews.

Low Fidelity

As part of the low fidelity, the *Design Mode* provides users with basic drawing features: pencil, eraser, ruler, image upload; control features: undo, redo, select; and collaboration features: share, comment. Users are provided with a drawing canvas in shape of the desired device (Figure 1). Additionally, users can zoom in/out to adjust the size of the canvas as per need. Users are also provided with an overview of all the sketched screens for quick access. The *Interaction Mode* (Figure 6) enables users to create, view, edit or delete four different kinds of interactions: screen transition, timer switch, URL, gesture based transition. The *Preview Mode* (Figure 7) can be used to preview the design and interactions of the LoFi prototype.

MetaMorph

MetaMorph is a UI element detector that identifies UI elements from a LoFi sketch. We built MetaMorph with a Deep Neural Network using TensorFlow Object Detection API [8]. Its detection model is RetinaNet [9] based Single-Shot MultiBox Detection (SSD) network with Resnet backbone [10]. It can identify 19 Google's Material Design based UI elements. It was evaluated with coco detection metrics and provides 84.98% mAP (mean Average Precision) with 72.66% AR (Average Recall).

To train MetaMorph, we generated a synthetic annotated dataset from UI element sketches. First, we collected UI element sketches from 350 participants, using paper and digital questionnaires, and created the UISketch dataset. This dataset contains 5906 sketches of 19 Google's Material Design based UI elements such as buttons, text fields, menus, etc. Later, we randomly sampled UI element sketches from UISketch dataset and placed them at random locations in a blank image to create a synthetic image. By this method, we generated a synthetic annotated dataset with 125000 images along with their respective XML, CSV, and TensorFlow record annotation files to train and evaluate MetaMorph.

Unlike Airbnb's sketching interfaces tool [15] and Microsoft's Sketch2Code [14], MetaMorph focuses on identifying UI elements from LoFi sketch instead of transforming it to code directly. We deployed MetaMorph as an API to make it a pluggable module which can be used independent of Eve. This API serves a swappable detection model which is scalable and can be retrained to detect other design language based UI element sketches. Its detection accuracy can also be fine-tuned further by retraining it with more UI element sketches.

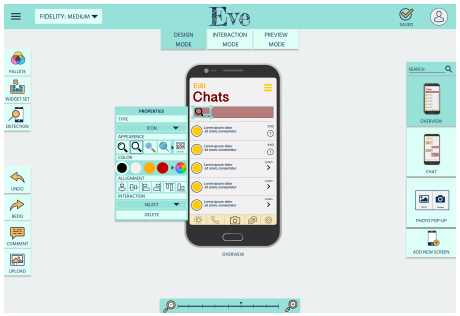

Figure 9: Eve: Properties of UI Element

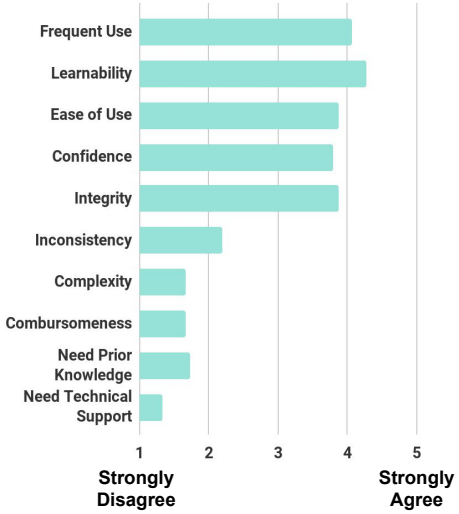

Figure 10: SUS Mean Responses

Medium Fidelity

The results of object detection are displayed as part of the MeFi *Design Mode*. Users can switch between viewing detected UI elements (Figure 8) and generated MeFi design (Figure 2). Unlike Microsoft's Sketch2Code [14] which directly converts LoFi to code, Eve enables users to control and monitor the conversion process. For each detected UI element, users can change its properties: type, appearance, color, alignment, interaction, etc. (Figure 9). In case a certain UI element remained undetected, users can also detect and label it manually. In addition to the screens overview, control and collaboration features, the design mode also provides users with the ability to choose the desired color palette and widget set. The *Interaction* and *Preview Mode* provide the same features as LoFi.

High Fidelity

Similar to MeFi, Eve automatically generates the HiFi for the sketched concept (Figure 3). In HiFi, the *Design Mode* additionally provides users with the feature to apply themes to the design. Finally, users can generate code for the application, executable on the corresponding platform. The *Interaction* and *Preview Mode* remain consistent for HiFi as well.

EVALUATION

As part of the usability evaluation, 15 UI/UX designers were asked to create the design and interactions of an Android chat application as LoFi, MeFi and HiFi prototype using Eve. All the 15 designers (F=6, M=9) were 23 to 35 years of age and had at least two years of prototyping experience. In addition to demographic information, they were asked to fill out the System Usability Scale (SUS) questionnaire [2]. Figure 10 shows the mean responses for each question. 87% of the participants said that they would like to use Eve frequently to create prototypes. 80% of the users found Eve easy to use. None of the users thought that they would require any technical support to use the system or that the system is unnecessarily complex. In summation, Eve scored an average of 78.5 points out of 100, which implies overall good usability.

SUMMARY & FUTURE WORK

Software prototyping involves the evolution of a concept into various stages of design such as low, medium and high fidelity prototypes. Initial workload analysis of software prototyping using NASA-TLX showed an increase in workload specifically in frustration, temporal demand, effort, and a decline in performance as the participants progressed from low to high fidelity. Upon reviewing 141 commercial and academic tools, we found that most tools look at prototyping as distinct steps, but in reality they are interconnected and therefore, require comprehensive tool support. In this paper, we introduced Eve; a sketch-based prototyping workbench that enables the users to sketch their ideas

as low fidelity prototypes while generating the consequent medium and high fidelity prototypes automatically. As per the usability evaluation using SUS, Eve scored an average of 78.5 points out of 100, which implies overall good usability. Based on these promising results, to further our research in future, we aim to investigate the impact of using Eve on subjective workload of software prototyping.

REFERENCES

[1] Tony Beltramelli. 2017. pix2code: Generating Code from a Graphical User Interface Screenshot. *CoRR* abs/1705.07962 (2017). arXiv:1705.07962 http://arxiv.org/abs/1705.07962

[2] John Brooke. 1996. SUS: A quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (jun 1996), 4–7. https://www.crcpress.com/product/isbn/9780748404605

[3] Anabela Caetano, Neri Goulart, Manuel Fonseca, and Joaquim Jorge. 2002. JavaSketchIt : Issues in Sketching the Look of User Interfaces. *Text* (2002), 9–14. http://www.aaai.org/Papers/Symposia/Spring/2002/SS-02-08/SS02-08-002.pdf

[4] Adrien Coyette, Stéphane Faulkner, Manuel Kolp, Quentin Limbourg, and Jean Vanderdonckt. 2004. SketchiXML. *Proceedings of the 3rd annual conference on Task models and diagrams - TAMODIA '04* (2004), 75. https://doi.org/10.1145/1045446.1045461

[5] Daniel Engelberg and Ahmed Seffa. 2002. A Framework for Rapid Mid-Fidelity Prototyping of Web Sites. (2002), 203–215. http://dl.acm.org/citation.cfm?id=646869.709393

[6] Brian Gore. 2010. Measuring and Evaluating Workload: A Primer. *NASA Technical Memorandum* July (2010), 35. https://matb-files.larc.nasa.gov/Workload_Primer_TM_Final.pdf

[7] Sandra. G. Hart and Lowell E. Staveland. 1988. Development of the NASA-TLX: Results of empirical and theoretical research. In *Human mental workload*, Peter A Hancock and Najmedin Meshkati (Eds.). Advances in Psychology, Vol. 52. North-Holland, 139–183. https://doi.org/10.1016/S0166-4115(08)62386-9

[8] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. 2016. Speed/accuracy trade-offs for modern convolutional object detectors. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* 2017-Janua (nov 2016), 3296–3305. https://doi.org/10.1109/CVPR.2017.351 arXiv:1611.10012

[9] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal Loss for Dense Object Detection. *CoRR* April (aug 2017), 79–84. arXiv:1708.02002 http://arxiv.org/abs/1708.02002

[10] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng Yang Fu, and Alexander C. Berg. 2016. SSD: Single shot multibox detector. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9905 LNCS (2016), 21–37. https://doi.org/10.1007/978-3-319-46448-0_2 arXiv:1512.02325

[11] Jorge Luis Perez Medina. 2016. The UsiSketch Software Architecture. *Romanian Journal of Human - Computer Interaction* 9, 4 (2016), 305–333. http://hdl.handle.net/2078.1/187342

[12] Beryl Plimmer and Mark Apperley. 2003. Software to sketch interface designs. *Human-Computer Interaction – INTERACT'03* (2003), 73–80. https://pdfs.semanticscholar.org/8211/6465daa1d31dd286657097339d3505459f5c.pdf

[13] Vinícius C. V. B. Segura, Simone D. J. Barbosa, and Fabiana Pedreira Simões. 2012. UISKEI: A Sketch-based Prototyping Tool for Defining and Evaluating User Interface Behavior. (2012), 18–25. https://doi.org/10.1145/2254556.2254564

[14] Spike Techniques, Microsoft, and Kabel. 2018. Sketch 2 Code. https://www.ailab.microsoft.com/experiments/30c61484-d081-4072-99d6-e132d362b99d

[15] Benjamin Wilkins. 2017. Sketching Interfaces - Airbnb Design. https://airbnb.design/sketching-interfaces/