Demo: HyperMind Builder – Pervasive User Interface to Create Intelligent Interactive Documents

Shoya Ishimaru
Nicolas Großmann
Andreas Dengel
German Research Center for Artificial Intelligence (DFKI)
Kaiserslautern, Germany
{firstname.lastname}@dfki.de

Carina Heisel
Pascal Klein
Jochen Kuhn
University of Kaiserslautern
Kaiserslautern, Germany
{lastname}@physik.uni-kl.de

Ko Watanabe
Yutaka Arakawa
Nara Institute of Science and Technology
Nara, Japan
{watanabe.ko.we2,ara}@is.naist.jp

UbiComp/ISWC'18 Adjunct, October 8–12, 2018, Singapore, Singapore
ACM 978-1-4503-5966-5/18/10.
https://doi.org/10.1145/3267305.3267667

Abstract
We introduce a Graphical User Interface (GUI) to create intelligent interactive documents for everyone. The intelligent interactive document refers to a document displaying contents dynamically according to a reader's behavior. To the best of our knowledge, creating such documents requires a certain amount of efforts and implementation skills. By utilizing our system, users including non-technological experts can create interactive documents without any programming. Our system supports many people to enhance the possibility of designing new human-document interactions.

Author Keywords
Graphical User Interface; Human-Document Interaction

ACM Classification Keywords
H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous

Introduction
Every human has different preferences in reading. For example, some people need details about a background for further understanding while the others do not require. It depends on who reads what. However, documents have traditionally been static. We believe that reading experiences should become more immersive and interesting if documents behave differently for each individual reader.

Figure 1: Overview of a work-flow in an educational scenario. Teachers create an interactive digital textbook, which displays information based on gaze (i.e., utilizing an eye tracker to measure the visual attention and to employ it for vivid interaction) for students.

Since the appearance of human sensing technologies including eye tracking, real-time collection of reading behavior has been getting more affordable in several environments [6, 5]. These technologies have enabled researchers to design intelligent interactive documents [4]. Furthermore, *Text 2.0* [1], a framework to create gaze-oriented dynamic documents in HTML and JavaScript, has helped software developers to implement interactive documents.

However, implementing interactions on documents is difficult especially for people who need it (e.g., teachers, publishers, and researchers in education). They still need helps of a person who can write programs to create interactive documents. In such a case, there is a possibility of having a discrepancy on the understanding of each other. It is difficult for teachers to explain their ideas completely, and software developers may misunderstand them.

In order to give everyone an opportunity to create intelligent interactive documents, we propose *HyperMind Builder* – GUI for creating intelligent interactive documents without requiring any programming skills. This paper presents an overview of our proposed system, application scenario, and an initial observation to investigate further improvements.

Approach

Figure 1 shows an overview of a work-flow of our proposed system. It consists of two processes: the creation and the activation of interactions.

Creating the Intelligent Interactive Document

We focus on designing a system with no programming and allowing intuitive operation for users. In our system, the screen is divided into three columns. In the central column, we provide an open source rich text editor. Since it is a WYSIWYG editor, a user can easily write texts and modify the styles or copy-and-paste texts from other shared contents. On each side of the editor, we arrange columns of material container. A user puts additional materials (e.g., images and videos related to the content) into the container by drag-and-drop. Providing columns on each side of the editor allows a user to add materials anywhere close to the context. After inserting materials, a user draws a hidden rectangle on the main content and creates relations between inserted materials and rectangles. After creating a document, a user can export and share the data with a format of *HyperMind Reader* [4]. In summary, our system requires only the writing of the content, drag-and-drop, and mouse clicks.

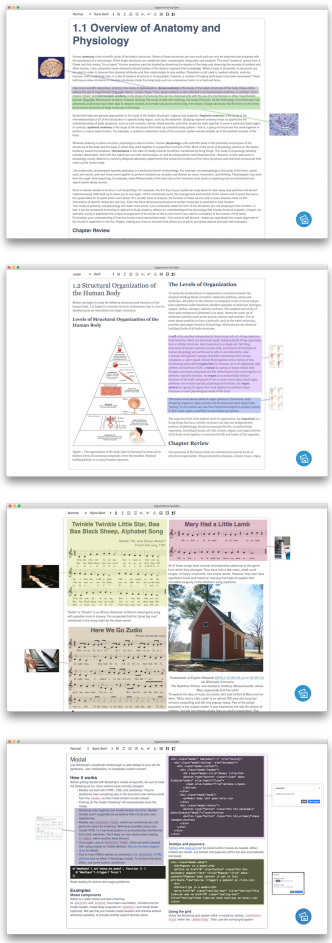

Figure 2: Examples of intelligent interactive documents created on our system. Document sources are from OpenStax, MILINE Library, and Twitter Bootstrap (CC license).

Activating the Intelligent Interactive Document

The additional materials around the main content will be displayed when they are required. The current implementation supports an activation based on a reader's attention. In other words, it utilizes an eye tracker and the related materials will be displayed if a reader's eye gaze is on a hidden rectangle longer than a threshold. We can register other activation rules (e.g., interest, comprehension, mental workload) if we utilize additional sensors.

Application Scenario

Figure 2 presents examples of documents created on our system. The most promising use-case is for textbooks. Additional materials which will be displayed when a student is interested in (or has troubles to understand) the content should improve the learning experience. In addition, the gaze-oriented interaction is useful in several scenarios including reading a musical score or programming codes.

Observation

In order to explore how simple and useful our system is for users, we have conducted a small study. The following section describes the condition and the analysis results.

Condition

We asked 10 college students with an age between 20 - 29 to participate in our study. We provided sample texts, supporting materials (videos and images), and multiple-choice questions related to the text for measuring the comprehension. Tasks for the participants were (1) to create an interactive document, (2) to read a document created by another participant, and (3) to solve multiple-choice questions. Before starting the tasks, we gave an instruction of our system to the participants. After the task, they answered surveys of NASA-TLX [3] and two free-writing questions.

Results and Discussion

Participants put 5.4 ± 2.1 supporting materials on a document. Figure 3 shows the result of NASA-TLX. From the result, we calculated weights by pair comparison on each factors as shown in Figure 4. This figure infers that higher the weight of factor, it corresponds to a cause of a high workload on each participant during the task. For instance, for Participant 9, *Performance* was the highest weight recorded. Hence, our system must improve *Performance* according to this participant. Overall result infer that *Mental Demand*, *Effort*, and *Temporal Demand* are factors that can be improved, and *Physical Demand* seems to be lower.

Regarding a free-writing question: "How was the usage of our system?", we obtained some answers like "I was a bit confused until I saw the example" or "Drag and drop were a bit confusing". We utilized the drag-and-drop because it is an intuitive function but we should consider preparing other options such as selecting from a list. But overall, we received several positive feedback including "It was really intuitive and useful" or "The usage of the system was straightforward and easy". We also asked participants about improvements: "Do you have any idea of the additional function for the system?" but there was any feedback related to the concept of the system.

Related Work

The closest concept to our system is the visual programming languages application like *Scratch* [7]. *Scratch* is an open-source media-rich programming environment. This application allowed many users to learn the concept of programming with an intuitive drag and drop method. It motivates many users and lowered the startup hurdle of programming. Our GUI toolkit has a similar concept, which is to lower the hurdle of creating an interactive document.

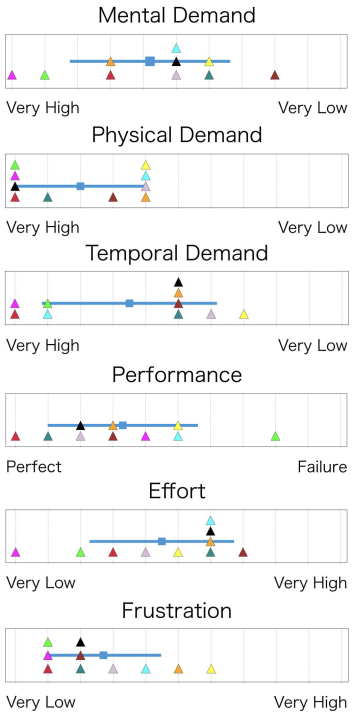

Figure 3: Result of NASA-TLX

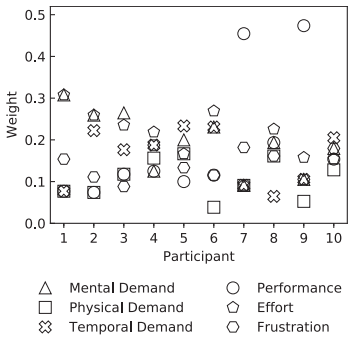

Figure 4: Weight of factors

Another similar concept is the idea proposed by Cheng et al. [2]. They come up with the idea of gazed-based gray shading annotation while reading. The aim of this proposal is to increase one's reading speed and understanding by guiding ones reading flow in a way how experts (teachers) read. As a result, they found out that annotation improved non-experts comprehension performance. Participants also felt gaze annotation as helpful. This study, authenticate the importance of the intelligent interactive document.

Conclusion and Future Work

We have implemented HyperMind Builder: GUI for to create intelligent interactive documents. From the observation, we overall proof the friendliness of a toolkit. Our next aim is to identify an effective activation rules for interactions. Thereby, we can add a new function of allowing creators to design not only what but also when and how supporting materials are be displayed.

Acknowledgements

HyperMind as a sub-project of "U.EDU: Unified Education – Medienbildung entlang der Lehrerbildungskette" is part of the "Qualitätsoffensive Lehrerbildung", a joint initiative of the Federal Government and the Länder which aims to improve the quality of teacher training. The program is funded by the Federal Ministry of Education and Research. This work was also partially supported by JSPS KAKENHI (16H01721)

REFERENCES

1. Ralf Biedert, Georg Buscher, Sven Schwarz, Manuel Möller, Andreas Dengel, and Thomas Lottermann. 2010. The text 2.0 framework: writing web-based gaze-controlled realtime applications quickly and easily. In *Proceedings of the 2010 workshop on Eye gaze in intelligent human machine interaction*. ACM, 114–117.
2. Shiwei Cheng, Zhiqiang Sun, Lingyun Sun, Kirsten Yee, and Anind K Dey. 2015. Gaze-based annotations for reading comprehension. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 1569–1572.
3. Sandra G Hart. 1986. NASA Task Load Index (TLX). Volume 1.0; Computerized Version. (1986).
4. Shoya Ishimaru, Syed Saqib Bukhari, Carina Heisel, Nicolas Großmann, Pascal Klein, Jochen Kuhn, and Andreas Dengel. 2018. Augmented Learning on Anticipating Textbooks with Eye Tracking. In *Positive Learning in the Age of Information*. Springer, 387–398.
5. Shoya Ishimaru, Soumy Jacob, Apurba Roy, Syed Saqib Bukhari, Carina Heisel, Nicolas Großmann, Michael Thees, Jochen Kuhn, and Andreas Dengel. 2017. Cognitive State Measurement on Learning Materials by Utilizing Eye Tracker and Thermal Camera. In *Proceedings of the 14th IAPR International Conference on Document Analysis and Recognition*, Vol. 8. IEEE, 32–36.
6. Kai Kunze, Susana Sanchez, Tilman Dingler, Olivier Augereau, Koichi Kise, Masahiko Inami, and Terada Tsutomu. 2015. The augmented narrative: toward estimating reader engagement. In *Proceedings of the 6th Augmented Human International Conference*. ACM, 163–164.
7. Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, and others. 2009. Scratch: programming for all. *Commun. ACM* 52, 11 (2009), 60–67.