Development of an Adaptive User Support System Based on Multimodal Large Language Models

Wei Wang*, Lin Li†, Shavindra Wickramathilaka*, John Grundy*, Hourieh Khalajzadeh‡, Humphrey O. Obie*
Anuradha Madugalla*
*Dept of Software Systems and Cybersecurity, Monash University, Melbourne, Australia
{wei.wang5, shavindra.wickramathilaka, john.grundy, humphrey.obie, anu.madugalla}@monash.edu
†Dept of Information Systems and Business Analytics, RMIT University, Melbourne, Australia
lin.li@rmit.edu.au
‡School of Information Technology, Deakin University, Melbourne, Australia
hkhalajzadeh@deakin.edu.au

***Abstract*—As software systems become more complex, some users find it challenging to use these tools efficiently, leading to frustration and decreased productivity. We tackle the shortcomings of conventional user support mechanisms in software and aim to create and assess a user support system that integrates Multimodal Large Language Models (MLLMs) for producing support messages. Our system initially segments the user interface to serve as a reference for selection and requests users to specify their preferences for support messages. Following this, the system creates personalised user support messages for each individual. We propose that user support systems enhanced with MLLMs can provide more efficient and bespoke assistance compared to conventional methods.**

***Index Terms*—adaptive user support, user interface, Multimodal Large Language Models (MLLMs)**

I. Introduction

The capabilities of Information Technology (IT) are advancing rapidly, yet the cognitive capabilities of users are not advancing at the same rate [1]. As software systems become increasingly complex, they pose significant challenges for many users to operate effectively and efficiently [2]. As a result, there is a growing need for end-user support, especially in aiding people from diverse backgrounds and skill levels to adjust to new platforms [3]. Historically, written text documentation has been the main method of providing user support in software systems (e.g., user manuals) [4], [5]. For beginners with minimal technical experience, mastering complex software through extensive written documentation still remains a daunting endeavour [6].

Research suggests that a smart guidance system that understands different user needs and offers relevant help can improve user experiences [1], [7], [8]. Large Language Models (LLMs) hold considerable promise in improving user support by interpreting the interface and offering customised assistance to various users. This includes understanding *key user differences*, *how users expect assistance to be provided* and *in what format they wish to receive their assistance* [3]. Although LLMs excel at handling and producing text, they lack proficiency in interpreting and creating content. MLLMs are LLM-based models capable of receiving, reasoning, and producing multimodal information. These models enhance traditional LLM functionalities by incorporating various data inputs, allowing them to understand and generate responses that integrate both text and visual elements [9], [10]. This research highlights the significance of adaptive user support in improving user satisfaction by leveraging MLLMs. By overcoming the limitations of traditional support methods, the goal is to create a user support system that integrates MLLMs for personalised help.

II. Related work

Traditional user support mechanisms require users to manually search large volumes of text, images, or tutorial videos to find the necessary information. This is often perceived by users as tedious and time-consuming [1], [11], [12]. There is substantial empirical evidence suggesting that traditional support mechanisms are less effective than anticipated [13], [14]. Given the remarkable capabilities of LLMs, these models are swiftly evolving, enabling novel forms of human-AI ***"cocreation"*** [15], [16]. Researchers have applied LLMs to a range of tasks in human-computer interaction and software engineering [17]–[21]. Two recent investigations have examined the potential of LLMs in the field of user support. Liu et al. [22] have developed a hint-text generation model that analyses interface data from input fields and uses in-context learning to generate hints that help visually impaired users understand data entry requirements. Babu et al. [3] employed LLMs to develop an effective recommendation system tailored for user guidance. Our study focus on the ***collaborative creation of user support messages*** in applications through the use of MLLMs.

III. Adaptive user support message generation

The adaptive user support system aims to offer contextually appropriate help to users by analysing their interactions with the user interface(UI)(Fig.1). The process starts with creating a UI segmentation prompt, which is examined by the MLLM model known as Language-based Interface Segmentation and Assistance (LISA) [23]. LISA segments the UI into different

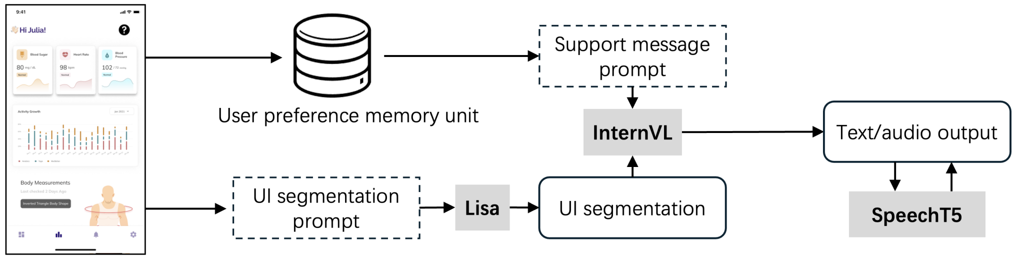

Fig. 1: Workflow of the adaptive support message generation

actionable components, which are then processed by the large-scale vision-language foundation model (InternVL) [24], tasked with managing visual-linguistic operations. Based on the segmented UI and stored user preferences, a support message prompt is created and fed back into the InternVL model to generate specific text outputs of support messages, providing guidance, instructions, or explanations that are contextually appropriate and meet user requirements. The generated text can be delivered audibly to users, especially aiding those with varying degrees of visual impairments. Moreover, users have the option to select particular UI elements, such as a health chart or an information component, to receive support or detailed, step-by-step guidance on key app features, including individual screens or elements. This adaptive approach ensures that users receive relevant and timely assistance, enhancing their overall experience with the system.

A *User Preference Memory Unit* is used to retain user preferences and past it to help ensure that support messages are tailored and contextually appropriate for each individual user. Without collecting or disrupting the software system, as MLLMs are capable of interpreting the UI as images, the tool can be applied universally across different applications. All user-related data will be gathered for individual users, ensuring that messages are personalised and consistent.

IV. UI Segmentation

The procedure starts with the application's UI, exemplified by a mHealth app shown in Fig.2(a), displaying various health metrics such as blood sugar, heart rate, blood pressure, activity growth, and body measurements. A text prompt is generated to request segmentation of different sections of the UI, *"Here is the application interface. Could you divide it into its respective sections for me?"* The LISA model is responsible for segmenting the UI into different components, producing a segmented UI version that emphasises the distinct sections identified for user assistance. In this segmented UI, various areas are marked to indicate where contextual guides can be offered (Fig.2(c)).

V. Support message

The source of the support message prompt comes from two origins. First, the system constructs the basic structure of the prompt for the segmented UI. Second, users themselves provide input on what the ***content*** of the support message should be and how they wish the message to be ***presented***. For example, users can specify whether they want additional instructions, detailed explanations, or a summary of the content. In a subsequent stage, they also have the option to convert the text output into audio output. This sophisticated method ensures that the support provided is comprehensive and tailored to the unique needs of each user. This is illustrated in Fig.2(b), where user prompts guide the system in creating more targeted and useful support messages.

After the support message prompt is generated, it is reintroduced into the InternVL model to produce specific text outputs for areas of the system that require additional context, as indicated by users through their interaction with UI image input. For instance, if the user is unsure about a particular feature or section of the interface, they can highlight this area, prompting the system to generate a detailed support message about it, as illustrated in Fig.2(c). The final output is a text-based support message that can be converted into audio as shown in Fig.2(d), assisting the user in navigating the UI by offering hints, instructions, or explanations tailored to the context and the user's needs. This adaptive approach ensures that users receive relevant and timely assistance, enhancing their overall experience with the system. Through the integration of system-generated and user-specific prompts, the adaptive user support system attains a significant degree of personalisation and efficiency in delivering user support.

VI. Future steps

We are implementing our adaptive support message generator with several use cases (e.g., mHealth applications). Enhancing the system's capacity to accommodate a broader range of interfaces and applications is essential. This will involve modifying existing models to seamlessly integrate with various software environments, including mobile apps, web platforms, and sophisticated enterprise systems. In addition, there is an opportunity to include multimodal user support that merges text, sound, and visual components. For instance, interactive videos can assist users in navigating complex tasks by pausing and emphasising various elements or screens, enabling users to follow along and grasp the context more effectively.

Acknowledgments

Wang, Wickramathilaka, Grundy and Madugalla are supported by ARC Laureate Fellowship FL190100035.

References

[1] A. Maedche, S. Morana, S. Schacht, D. Werth, and J. Krumeich, "Advanced user assistance systems," *Business & Information Systems Engineering*, vol. 58, pp. 367–370, 2016.

[2] M. J. Albers, "Design and usability: Beginner interactions with complex software," *Journal of technical writing and communication*, vol. 41, no. 3, pp. 271–287, 2011.

[3] S. K. Babu, M. Chetitah, and S. von Mammen, "Recommender-based user guidance framework," in *2024 IEEE International Conference on Artificial Intelligence and eXtended and Virtual Reality (AIxVR)*, pp. 275–280, IEEE, 2024.

[4] E. Aghajani, C. Nagy, O. L. Vega-Márquez, M. Linares-Vásquez, L. Moreno, G. Bavota, and M. Lanza, "Software documentation issues unveiled," in *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, pp. 1199–1210, IEEE, 2019.

[5] E. Aghajani, C. Nagy, M. Linares-Vásquez, L. Moreno, G. Bavota, M. Lanza, and D. C. Shepherd, "Software documentation: the practitioners' perspective," in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, pp. 590–601, 2020.

[6] K. Arning, S. Himmel, and M. Ziefle, "You can ('t) teach an old dog new tricks: analyzing the learnability of manufacturing software systems in older users," in *Human Aspects of IT for the Aged Population. Healthy and Active Aging: Second International Conference, ITAP 2016, Held as Part of HCI International 2016 Toronto, ON, Canada, July 17–22, 2016, Proceedings, Part II 2*, pp. 277–288, Springer, 2016.

[7] D. Kao, "Exploring help facilities in game-making software," in *Proceedings of the 15th International Conference on the Foundations of Digital Games*, pp. 1–14, 2020.

[8] M. Wooldridge and N. R. Jennings, "Intelligent agents: Theory and practice," *The knowledge engineering review*, vol. 10, no. 2, pp. 115–152, 1995.

[9] S. Yin, C. Fu, S. Zhao, K. Li, X. Sun, T. Xu, and E. Chen, "A survey on multimodal large language models," *arXiv preprint arXiv:2306.13549*, 2023.

[10] J. Wu, W. Gan, Z. Chen, S. Wan, and S. Y. Philip, "Multimodal large language models: A survey," in *2023 IEEE International Conference on Big Data (BigData)*, pp. 2247–2256, IEEE, 2023.

[11] A. Shachak, R. Dow, J. Barnsley, K. Tu, S. Domb, A. R. Jadad, and L. Lemieux-Charles, "User manuals for a primary care electronic medical record system: A mixed-methods study of user-and vendor-generated documents," *IEEE transactions on professional communication*, vol. 56, no. 3, pp. 194–209, 2013.

[12] G. Veletsianos, "Cognitive and affective benefits of an animated pedagogical agent: Considering contextual relevance and aesthetics," *Journal of Educational Computing Research*, vol. 36, no. 4, pp. 373–377, 2007.

[13] T. A. Sykes, "Support structures and their impacts on employee outcomes," *MIS quarterly*, vol. 39, no. 2, pp. 473–496, 2015.

[14] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner, "Three decades of driver assistance systems: Review and future perspectives," *IEEE Intelligent transportation systems magazine*, vol. 6, no. 4, pp. 6–22, 2014.

[15] N. Davis, C.-P. Hsiao, Y. Popova, and B. Magerko, "An enactive model of creativity for computational collaboration and co-creation," *Creativity in the digital age*, pp. 109–133, 2015.

[16] S. Khan, "Harnessing gpt-4 so that all students benefit. a nonprofit approach for equal access," Nov. 2023.

[17] H. Pearce, B. Tan, B. Ahmad, R. Karri, and B. Dolan-Gavitt, "Examining zero-shot vulnerability repair with large language models," in *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 2339–2356, IEEE, 2023.

[18] F. Gmeiner, J. L. Conlin, E. H. Tang, N. Martelaro, and K. Holstein, "An evidence-based workflow for studying and designing learning supports for human-ai co-creation," in *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pp. 1–15, 2024.

[19] T. Ahmed and P. Devanbu, "Better patching using llm prompting, via self-consistency," in *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pp. 1742–1746, IEEE, 2023.

[20] D. Russo, "Navigating the complexity of generative ai adoption in software engineering," *ACM Transactions on Software Engineering and Methodology*, 2024.

[21] C. Y. Kim, C. P. Lee, and B. Mutlu, "Understanding large-language model (LLM)-powered human-robot interaction," in *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 371–380, 2024.

[22] Z. Liu, C. Chen, J. Wang, M. Chen, B. Wu, Y. Huang, J. Hu, and Q. Wang, "Unblind text inputs: Predicting hint-text of text input in mobile apps via llm," in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI '24, ACM, May 2024.

[23] X. Lai, Z. Tian, Y. Chen, Y. Li, Y. Yuan, S. Liu, and J. Jia, "Lisa: Reasoning segmentation via large language model," *arXiv preprint arXiv:2308.00692*, 2023.

[24] Z. Chen, J. Wu, W. Wang, W. Su, G. Chen, S. Xing, Z. Muyan, Q. Zhang, X. Zhu, L. Lu, *et al.*, "Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks," *arXiv preprint arXiv:2312.14238*, 2023.

APPENDIX

Fig. 2: Prototype of the adaptive support message generation process