

# Exploring the Role of End Users in Performing EUD with Large Language Models

Luigi Gargioni<sup>1,\*</sup>, Barbara Rita Barricelli<sup>1</sup>, Daniela Fogli<sup>1</sup> and Angela Locoro<sup>2</sup>

<sup>1</sup>Department of Information Engineering, University of Brescia, Brescia, Italy

<sup>2</sup>Department of Economics and Management, University of Brescia, Brescia, Italy

## Abstract

Large Language Models (LLMs) are being used to expand the concept of End-User Development (EUD), allowing end users to describe their needs related to the creation, modification, extension or testing of digital artifacts in natural language. This paper presents a survey on recent papers that explore the integration of EUD with LLMs. The final aim is to reflect on the opportunities offered by LLMs to EUD and on the challenges to address, to understand how to empower end users rather than diminish their role in tailoring systems.

## Keywords

End-User Development, Meta-design, Large Language Model, Literature review

## 1. Introduction

A first definition of End-User Development (EUD) was proposed in 2006 by the related European Network of Excellence conceiving it as “a set of methods, techniques, and tools that allow users of software systems, who are acting as non-professional software developers, at some point to create, modify, or extend a software artifact” [1]. This concept evolved over the years, due to the introduction of new technologies (e.g., Internet of Things, collaborative and social robots, virtual assistants, immersive video games, and so on) that can be tailored by end users to obtain a desired behavior. In addition, also the characteristics of end users performing EUD changed over the years, since in the past they were mainly domain experts working in an organization and possessing the very knowledge for system creation and adaptation; today, they can be lay users who would like to tailor the behavior of their smart home or virtual assistant (e.g., Amazon Alexa, Google Assistant or Apple Siri). Hence, the definition proposed in [2] helps widening the perspective on both EUD and end users. Here, EUD is regarded as “the set of methods, techniques, tools, and socio-technical environments that allow end users to act as professionals in those ICT-related domains in which they are not professionals, by creating, modifying, extending and testing digital artifacts without requiring knowledge in traditional software engineering techniques”. It is

---

*Proceedings of the 8th International Workshop on Cultures of Participation in the Digital Age (CoPDA 2024): Differentiating and Deepening the Concept of “End User” in the Digital Age, June 2024, Arenzano, Italy*

\*Corresponding author.

✉ luigi.gargioni@unibs.it (L. Gargioni); barbara.barricelli@unibs.it (B. R. Barricelli); daniela.fogli@unibs.it (D. Fogli); angela.locoro@unibs.it (A. Locoro)

🆔 0000-0003-4354-916X (L. Gargioni); 0000-0001-9575-5542 (B. R. Barricelli); 0000-0003-1479-2240 (D. Fogli); 0000-0002-6740-8620 (A. Locoro)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

worth noticing that this definition considers digital artifacts in general, rather than just software artifacts, and regards end users as acting as professionals in ICT-related domains, rather than as just non-professional software developers. This opens up different nuances for EUD activities and types of end users. Ye and Fischer anticipated this aspect in 2007 [3], pointing out how the distinction between users and developers is going to disappear with the time passing. While, almost fifteen years before, Nardi [4] claimed that the end user is *“the person who does not want to turn a task into a programming problem, who would rather follow a lengthy but well-known set of procedures to get the job done”*.

One of the most important features of EUD environments is their capability to support end users to create, modify, or extend digital artifacts following their reasoning habits [5, 6] thanks to the adoption of familiar languages and direct mappings with real-world objects. In this context, Large Language Models (LLMs) promise to push further the concept of End-User Development, enabling end users to describe their EUD activities in natural language, rather than interacting with graphical interfaces. Several researchers are currently investigating this topic to provide end users with novel approaches to performing EUD.

This paper analyzes recent published papers found through Google Scholar on the relationship between “End-User development” and “Large Language Models”. The aim is to investigate in which EUD applications an LLM-based approach has been applied, and who are the target end users of these systems. The goal is to identify not only the opportunities offered by LLMs to EUD, but also to understand how the role of end users may change due to LLMs, and to explore the limitations affecting LLM-based EUD environments.

## **2. Survey on EUD supported by LLMs**

### **2.1. Methodology**

The survey reported in this paper aims to answer the following Research Question (RQ): *“How does recent scientific literature consider the relationship between Large Language Models and End-User Development?”*.

To include as many recent results as possible in our survey, we used Google Scholar with the query: (“End-User Development” OR “EUD”) AND (“Large Language Model” OR “LLM”). The query was performed on February 3rd 2024, and looked for papers published since 2023, considering that the studies on the relationship between EUD and LLMs started to be performed only after the availability of OpenAI ChatGPT in November 2022.

Four researchers conducted an independent screening of the title and venue of the 103 papers found through the query, in order to assess their eligibility for further reading and analysis. In case of discrepancies, the reviewers resolved them through discussion. 83 papers were excluded in this phase, due to one of the following criteria: out-of-scope research, duplicate paper, pre-print, conference description paper, special issue editorial, not in English. The full-text of the remaining 20 papers was then read independently by the researchers to perform a further selection. Nine papers were finally selected since they were considered relevant to our RQ.

## 2.2. Selected papers

The selected papers have been classified in the following categories: i) theoretical papers, ii) papers discussing code generation through LLMs, iii) papers addressing the creation of automations for smart environments, iv) papers focused on image generation for design.

***Theoretical papers.*** Papers [7] and [8] explore the relationship between EUD and LLMs from a theoretical perspective. After a description of the main characteristics of adaptive systems (based on AI, including ChatGPT) and adaptable systems (based on EUD techniques), Fischer [7] proposes to design solutions that create a symbiosis between adaptive and adaptable systems. A conversational approach is advocated when LLMs are used, since the answers generated by ChatGPT (or other LLMs) can be considered most valuable when seen as starting points and inspirations for users, who may subsequently adapt and improve the generated output (also through additional prompting). In general, the benefits of both types of systems must be strengthened; but, more importantly, drawbacks of adaptive systems (e.g., the lack of explainability, privacy intrusions) and of adaptable systems (e.g., the participation overload, the incompatibility among different modified versions) must be addressed pushing investigations about the relationship between AI and EUD beyond technological issues, deepening users' motivation, control, ownership, and autonomy.

Repenning and Grabowski [8], instead, re-frame Computational Thinking as a prompting engineering activity that may empower users in creating programs (e.g., software simulations to be used in the education context) or contents (e.g., new images). As to the first example, ChatGPT can be used by a teacher to enrich their lessons: a step-by-step refinement process of prompts is recognized to be necessary to reach the desired output, but a drawback is identified when the Javascript program generated by ChatGPT to simulate a specific phenomenon does not behave as requested: in this case, the teacher might not be able to fix the program code. Similarly, the use of DALL-E to generate images is evaluated; several conversation turns (prompts) are needed also in this case to achieve the user's objective. However, it emerges that it can be difficult to understand system output, due to the unpredictable nature of the underlying AI, or it may be frustrating continuing with new prompts when the generated output is far from the desired one, as in case of geometric pictures. An interesting aspect highlighted in this work is the necessity for users to acquire new competencies concerning the skills to pose questions and to modify them on system's reply, namely *prompting*. This implies that future EUD environments should help users develop such skills and provide suggestions on how to specify prompts.

***Papers on LLMs supporting code generation.*** Four out of 9 papers discuss the integration of LLMs in EUD environments to support users in code generation. The paper [9] presents a user-oriented approach that employs LLMs to enable end users to create websites through natural language specifications. The method uses prompt engineering to ensure LLM response adherence to a specific template, allowing for direct parsing of the input. The approach enables users to refine the generated website without worrying about the underlying code by using the model's responses. A proof-of-concept implementation using GPT-4 is presented. Finally, the paper discusses future research directions. These include integrating the approach with low-code/no-code platforms and conducting user studies to assess its efficacy and utility. The

ultimate goal is to democratize website development and make it more accessible to users without technical expertise.

In [10], the authors investigate the integration of ChatGPT in a EUD environment for collaborative robot programming. Collaborative robots are increasingly being used across various industries due to their ability to work alongside human workers. However, the complexity of programming tasks for these robots can be a barrier to their widespread adoption. To address this challenge, an intuitive environment, called CAPIRCI, which combines natural language interaction with a chat-based interface and visual interaction with a block-based graphic interface, was proposed in [11, 12]. In [10], the authors investigate the use of ChatGPT in CAPIRCI to generate XML descriptions of robot tasks based on user requests expressed in natural language, which can be translated into a block-based representation of the robot program. This representation can be visually manipulated to ensure correctness and completeness. The proposed approach could advance EUD for robot programming, especially for users with limited computational fluency.

The code verification topic is the focus of a further selected paper: [13] examines the errors commonly made by LLMs in robot programming. These errors are categorized into interpretation and execution phases, with a particular focus on execution errors caused by LLMs' forgetfulness of key information provided in user prompts. The authors suggest implementing prompt engineering tactics to decrease execution errors. They demonstrate these tactics using three models: ChatGPT, Bard, and LLaMA-2. The paper provides practical strategies for error mitigation, such as reinforcing task constraints in the objective prompt and storing numerical task contexts in data structures. It concludes by emphasising the necessity of a set of tools for the productive use of LLM-based robot programming. These tools may include custom verification scripts and a preview tool to simulate program behavior before robot deployment.

Finally, paper [14] presents the result of a study about the impact of AI code generators on introductory programming for novice programmers aged 10-17. Specifically, the authors performed a controlled experiment involving 69 novices who were asked to perform 45 code-authoring tasks in Python. Half of them were asked to use OpenAI Codex to generate the code automatically. The experiment demonstrated that novice programmers using AI code generators performed better, faster, and with less frustration and did not encounter problems in modifying the code subsequently.

***Papers on LLMs supporting the creation of automations.*** Two papers consider the use of LLMs to facilitate the creation of automations (i.e., trigger-action rules) for a IoT ecosystem like a smart home. The paper [15] focuses on enhancing user interaction with smart home automation through augmented reality (AR) and explainable artificial intelligence (XAI). It presents a mobile AR platform called ARACS (Augmented Reality Automation Creation and Simulation) that applies the XAIR (Explainable AI in Augmented Reality) framework to interact with the system. ARACS is an Android application that uses AR to overlay information on physical objects in the user's environment, allowing users to dynamically configure automation rules by interacting with visualizations placed over objects. The role of the LLM is to provide descriptions of the context representation and its embeddings with natural language generated by BERT <sup>1</sup>. The

---

<sup>1</sup>We decided to keep the point of the authors, who claim that BERT is an LLM, although this inclusion is controversial.

**Table 1**

The papers included in the study.

Ref.	End Users	Output	LLMs	Interaction
[7]	Domain experts	–	ChatGPT	Conversational
[8]	Non-experts	Javascript code, images	ChatGPT, DALL-E	Conversational
[9]	Non-experts	HTML+CSS	ChatGPT	Conversational
[10]	Domain experts	Robot task	ChatGPT	Conversational
[13]	Expert programmers	Robot task	ChatGPT, Bard, LLaMa-2	One-Shot
[14]	Novice programmers	Python code	OpenAI Codex	One-Shot
[15]	Non-experts	IoT rules	BERT	One-Shot
[16]	Non-experts	IoT rules	ChatGPT	Conversational
[17]	Architecture students	Images	Dall-E, Stable Diffusion	Conversational

paper discusses two scenarios to illustrate the application of the XAIR framework to the ARACS platform. The first scenario involved rule recommendations during automation creation. The second scenario used the environment simulator/debugger of the application to understand and fix automation issues, with explanations triggered automatically to anticipate a possible user’s misunderstanding.

In [16], a conversational natural-language-based system for creating rules to control elements in an IoT ecosystem is presented. The system architecture integrates ChatGPT and Rasa (an open-source framework for chatbot development). Specifically, ChatGPT components are used to split complex rules, manage breakdowns, and answer questions, while Rasa handles intent recognition and entity management, extraction, and conversational flow management. The system is designed to manage complex inputs that describe rules consisting of multiple triggers and actions.

**Papers on LLMs supporting design.** The paper [17] investigates the influence of Large-scale Text-to-image Generative Models (LTGMs) on creativity, focusing on tools like DALL-E, MidJourney, and StableDiffusion. The study aims to analyze feedback from design students working on architectural projects during a workshop. The participants were asked to design a public repository of water, with the aim to prioritize a rich sensory experience and spatial configuration. After the design task with LTGMs, they were asked to respond to a questionnaire to gather their experiences, challenges, and the perceived potential and weaknesses of these models in the creative process. The results of the study highlighted that, although the students appreciated the tools for their ability to produce a variety of interesting images, which could enhance the initial stages of design, they failed to be in control of the output. The paper emphasizes that future development should focus on making LTGMs more interactive and transparent, enabling users to customize their use to fit creative needs better.

### 3. Discussion and Conclusion

The list of selected papers is provided in Table 1, which includes additional information related to the four dimensions we used to classify them: the target end users who are meant to use the

described systems, the LLM-generated output obtained by their use, the LLMs adopted, and the type of interaction that the systems provide.

Most of the papers focus on facilitating code generation, suggesting that end users should be regarded as novice programmers. However, end users traditionally targeted by EUD are domain experts who are interested in doing their job more effectively and efficiently, but might not possess the adequate programming knowledge to verify if the generated code satisfies their needs. Even though some of the papers recognize this problem and propose solutions to it, one may consider whether end users should acquire in the next future competencies in software development or specific Computational Thinking (CT) skills that help them in this matter. Alternatively, EUD environments based on LLMs should be designed considering their EUDability in comparison with the CT skills of their users [18, 19].

LLMs are known to technical users to sometimes produce inaccurate results due to hallucinations. It is important to note that end users may initially blindly trust the answers provided by LLMs, assuming they are correct, without being aware of their probabilistic and non-deterministic nature. Trust in technology drops only after realizing that the output is not always correct. This problem raises the question of how users can verify the proposed output. Verification methodologies for the final output are essential, as illustrated in the analyzed papers. For instance, [10] suggests providing users with a block-based representation of the defined robot task for final confirmation; [9] proposes reaching a final result through successive refinements by a conversation flow displaying the proposed web interface each time; finally, in [13] the goal is to address the problem of hallucinations by identifying the tasks that these systems most frequently fail on, in order to formulate more precise prompts to compose robot tasks. The effectiveness of LLMs is undoubtedly impressive, but it is still crucial to involve humans in the development process to ensure the accuracy of the results.

A further problem emerging from the use of LLMs in creative design is the lack of support to control intermediate communication/adjustment interventions during the generative production of artifacts. This limitation may hinder the benefits of trusting LLMs for a profitable teaming. To overcome this problem, we need to find a trade-off between the maximization of design creativity in prompt generation and the realization of step-wise and improved explainability protocols for supporting subjects' critical thinking while co-creating with the machine.

In synthesis, future EUD environments integrating LLMs should incorporate much more social features than today. In particular, a conversational approach, based on step-by-step refinements, should be preferred to one-shot interaction, paying attention to enabling users control the conversation. Another aspect is related to providing end users with clear instructions, tutorials, and other shared materials on how to pose questions and modify system replies. Finally, EUD environments must include tools for checking correctness easily, e.g. with simulators or sandboxes in case of code generation. In this way, we can regard LLM-based EUD as empowering end users rather than diminishing or substituting them. This requires adopting meta-design approaches [20, 21] that consider both social and technical issues related to the design of EUD environments that are not only easy to use but also transparent, trustworthy, and reliable [22].

## References

- [1] H. Lieberman, F. Paternò, V. Wulf, *End User Development (Human-Computer Interaction Series)*, Springer-Verlag, Berlin, Heidelberg, 2006.
- [2] B. R. Barricelli, F. Cassano, D. Fogli, A. Piccinno, End-user development, end-user programming and end-user software engineering: A systematic mapping study, *Journal of Systems and Software* 149 (2019) 101–137. doi:<https://doi.org/10.1016/j.jss.2018.11.041>.
- [3] Y. Ye, G. Fischer, Designing for participation in socio-technical software systems, in: C. Stephanidis (Ed.), *Universal Access in Human Computer Interaction. Coping with Diversity*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 312–321.
- [4] B. A. Nardi, *A Small Matter of Programming: Perspectives on End User Computing*, MIT Press, Cambridge, MA, USA, 1993.
- [5] A. Repenning, A. Ioannidou, What makes end-user development tick? 13 design guidelines, in: *End User Development*, Springer Netherlands, Dordrecht, 2006, pp. 51–85. doi:[10.1007/1-4020-5386-X\\_4](https://doi.org/10.1007/1-4020-5386-X_4).
- [6] K. Whiley, A. F. Blackwell, Visual programming in the wild: A survey of labview programmers, *Journal of Visual Languages and Computing* 12 (2001) 435–472. doi:<https://doi.org/10.1006/jv1c.2000.0198>.
- [7] G. Fischer, Adaptive and adaptable systems: Differentiating and integrating ai and eud, in: *End-User Development: 9th International Symposium, IS-EUD 2023*, Cagliari, Italy, June 6–8, 2023, Proceedings, Springer-Verlag, Berlin, Heidelberg, 2023, p. 3–18. doi:[10.1007/978-3-031-34433-6\\_1](https://doi.org/10.1007/978-3-031-34433-6_1).
- [8] A. Repenning, S. Grabowski, Proomting is computational thinking, in: A. Bellucci, L. D. Russis, P. Díaz, A. I. Mørch, D. Fogli, F. Paternò (Eds.), *Joint Proceedings of the Workshops, Work in Progress Demos and Doctoral Consortium at the IS-EUD 2023 co-located with the 9th International Symposium on End-User Development (IS-EUD 2023)*, Cagliari, Italy, June 6-8, 2023, volume 3408 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023, pp. 1–6. URL: <https://ceur-ws.org/Vol-3408/short-s2-07.pdf>.
- [9] T. Calò, L. De Russis, Leveraging large language models for end-user website generation, in: *End-User Development: 9th International Symposium, IS-EUD 2023*, Cagliari, Italy, June 6–8, 2023, Proceedings, Springer-Verlag, Berlin, Heidelberg, 2023, p. 52–61. doi:[10.1007/978-3-031-34433-6\\_4](https://doi.org/10.1007/978-3-031-34433-6_4).
- [10] G. Bimbatti, D. Fogli, L. Gargioni, Can chatgpt support end-user development of robot programs?, in: A. Bellucci, L. D. Russis, P. Díaz, A. I. Mørch, D. Fogli, F. Paternò (Eds.), *Joint Proceedings of the Workshops, Work in Progress Demos and Doctoral Consortium at the IS-EUD 2023 co-located with the 9th International Symposium on End-User Development (IS-EUD 2023)*, Cagliari, Italy, June 6-8, 2023, volume 3408 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023, pp. 1–8. URL: <https://ceur-ws.org/Vol-3408/short-s2-03.pdf>.
- [11] S. Beschi, D. Fogli, F. Tampalini, Capirci: a multi-modal system for collaborative robot programming, in: *End-User Development: 7th International Symposium, IS-EUD 2019*, Hatfield, UK, July 10–12, 2019, Proceedings 7, Springer, 2019, pp. 51–66.
- [12] D. Fogli, L. Gargioni, G. Guida, F. Tampalini, A hybrid approach to user-oriented programming of collaborative robots, *Robotics and Computer-Integrated Manufacturing* 73 (2022) 102234.

- [13] J.-T. Chen, C.-M. Huang, Forgetful large language models: Lessons learned from using llms in robot programming, 2023. [arXiv:2310.06646](https://arxiv.org/abs/2310.06646).
- [14] M. Kazemitabaar, J. Chow, C. K. T. Ma, B. J. Ericson, D. Weintrop, T. Grossman, Studying the effect of ai code generators on supporting novice learners in introductory programming, in: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, CHI '23, Association for Computing Machinery, New York, NY, USA, 2023, pp. 1–23. doi:10.1145/3544548.3580919.
- [15] A. Mattioli, F. Paternò, Towards explainable automations in smart homes using mobile augmented reality, in: A. Bellucci, L. D. Russis, P. Díaz, A. I. Mørch, D. Fogli, F. Paternò (Eds.), Joint Proceedings of the Workshops, Work in Progress Demos and Doctoral Consortium at the IS-EUD 2023 co-located with the 9th International Symposium on End-User Development (IS-EUD 2023), Cagliari, Italy, June 6-8, 2023, volume 3408 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023, pp. 1–7. URL: <https://ceur-ws.org/Vol-3408/short-s4-07.pdf>.
- [16] S. Gallo, A. Malizia, F. Paternò, Towards a chatbot for creating trigger-action rules based on chatgpt and rasa, in: A. Bellucci, L. D. Russis, P. Díaz, A. I. Mørch, D. Fogli, F. Paternò (Eds.), Joint Proceedings of the Workshops, Work in Progress Demos and Doctoral Consortium at the IS-EUD 2023 co-located with the 9th International Symposium on End-User Development (IS-EUD 2023), Cagliari, Italy, June 6-8, 2023, volume 3408 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023, pp. 1–6. URL: <https://ceur-ws.org/Vol-3408/short-s4-01.pdf>.
- [17] T. Turchi, S. Carta, L. Ambrosini, A. Malizia, Human-ai co-creation: evaluating the impact of large-scale text-to-image generative models on the creative process, in: International Symposium on End User Development, Springer, 2023, pp. 35–51.
- [18] B. R. Barricelli, D. Fogli, A. Locoro, Eudability: A new construct at the intersection of end-user development and computational thinking, *Journal of Systems and Software* 195 (2023) 111516. doi:10.1016/j.jss.2022.111516.
- [19] B. R. Barricelli, D. Fogli, A. Locoro, Designing for a sustainable digital transformation: The dea methodology, in: L. D. Spano, A. Schmidt, C. Santoro, S. Stumpf (Eds.), *End-User Development*, Springer Nature Switzerland, Cham, 2023, pp. 189–199.
- [20] G. Fischer, E. Giaccardi, Meta-design: A framework for the future of end-user development, in: H. Lieberman, F. Paternò, V. Wulf (Eds.), *End User Development*, Springer Netherlands, Dordrecht, 2006, pp. 427–457. doi:10.1007/1-4020-5386-X\_19.
- [21] G. Fischer, D. Fogli, A. Piccinno, Revisiting and broadening the meta-design framework for end-user development, in: F. Paternò, V. Wulf (Eds.), *New Perspectives in End-User Development*, Springer International Publishing, Cham, 2017, pp. 61–97. doi:10.1007/978-3-319-60291-2\_4.
- [22] B. Shneiderman, *Human-Centered AI*, Oxford University Press, Oxford, UK, 2022.