

Adaptive and Adaptable Systems: Differentiating and Integrating AI and EUD

Gerhard Fischer

Center for LifeLong Learning & Design (L3D)
Department of Computer Science and Institute of Cognitive Science
University of Colorado, Boulder USA
gerhard@colorado.edu

Abstract. The framework presented in the paper identifies the promises and pitfalls of *Artificial Intelligence (AI)* and *End-User Development (EUD)* approaches by focusing on two basic system components: (1) *adaptive systems* (grounded in AI) that change their behavior automatically driven by context-aware mechanisms including models of their users and specific task contexts, and (2) *adaptable systems* (grounded in EUD) that can be adjusted, modified, and extended by their users in order to capture unforeseen and important emergent user needs and aspects of problems. Grounded in an analysis of design trade-offs between the two approaches, arguments, and examples for creating a desirable symbiosis between adaptive and adaptable systems are described and design guidelines for future socio-technical environments are explored contributing to the development of theoretical concepts for the future of EUD.

Keywords: artificial intelligence, adaptive systems, context-aware interactions, user and task modeling, personalization, end-user development, adaptable systems meta-design, creativity, Auto-Correct, ChatGPT, design trade-offs, design guidelines

1 Introduction

Research efforts arguing for the need and desirability of socio-technical environments that support users of all ages to think, learn, work, and collaborate in more productive and more creative ways have been the objective of numerous research disciplines including computer science, cognitive science, and the learning sciences [1]. More specifically, the research has been pursued in *Artificial Intelligence (AI)* [2] and *End-User Development (EUD)* [3]. AI and EU are research areas which have multiple meanings and are often used without any detailed descriptions and simple definitions of the two research themes are not sufficiently descriptive. By focusing on *adaptive* approaches (relying primarily on AI) and *adaptable* approaches (relying primarily on EUD), the paper attempts to deepen the understanding of the challenges, promises, pitfalls, and design trade-offs associated with future socio-technical environments contributing to the *quality of life* in the digital age [4-6].

2 Differentiating AI and EUD approaches

There is no generally accepted definition for AI and there is no defined boundary to separate “AI systems” from “non-AI systems”. Despite this shortcoming AI is currently being considered world-wide as a “Deus ex Machina” and it is credited with miraculous abilities to solve all problems and exploit unique opportunities of the digital age (the most prominent example emerging in late 2022 being ChatGPT that is briefly discussed later in the paper). AI (as characterized by this paper) explores technology-centered approaches, replacing human beings by automating how decisions and conclusions are reached. AI involves the creation of models and algorithms that can analyze data, learn from it, and make predictions or decisions based on that data. These kinds of AI applications include natural language processing, image recognition, self-driving vehicles, and chatbots.

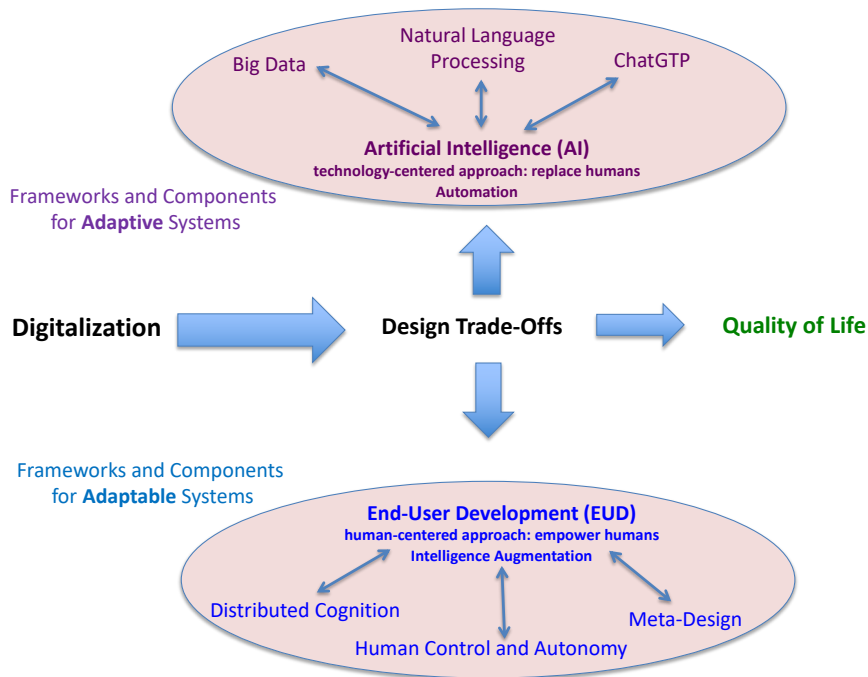


Fig. 1. Differentiating AI and EUD Approaches.

EUD is instrumental for the ability to create and reformulate knowledge, for allowing all stakeholders to express themselves creatively and appropriately, and for producing and generating information rather than simply comprehending existing information. It supports diverse audiences in designing and building their own artifacts by situating computation in new contexts, by generating content, and by developing tools that democratize design, innovation, and knowledge creation [7]. With computers and software becoming pervasive, many domain experts have started to develop or

adapt sophisticated software systems as an integral part of their work to fully utilize the power of the computer. They are not professionally educated as software engineers but spend a great deal of their time creating software systems for their own work. Given how domain experts' needs, goals, and education differ from those of professional software engineers, end-user development research should not be based on a scaled-down version of, or a simple transfer from, current software engineering principles [8].

Figure 1 tries to illustrate the relationship between the two approaches.

3 AutoCorrect: A Simple Example for Illustration

The following simple example illustrates the abstract approaches in a specific context. AutoCorrect is a software feature commonly found in word processing programs (such as Microsoft Word). The adaptive part of AutoCorrect detects and corrects misspelled words e.g., (1) “hte” is transformed into “the” and (2) “EHR” into “HER”.

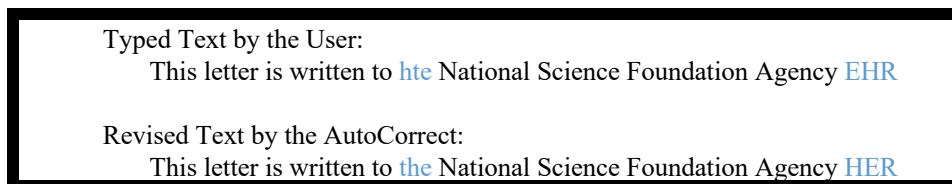


Fig. 2. The adaptive component of AutoCorrect in action.

These modifications are done *automatically* by an AI-based systems relying on 1) an extensive dictionary of correct spellings in the English language, (2) the recognition that “hte” or “EHR” are no English words, and (3) the knowledge that transposition errors are mistakes that people make frequently.

The example shows the fundamental limitations of adaptive features (and AI systems in general): their knowledge of the world is limited. While the change of “hte” to “the” in Figure 2 is welcome, the replacement of “EHR” is inappropriate in this context. The example documents a real event: by going unnoticed in a communication with the NSF funding agency “Education and Human Resources (EHR)”, the change (being unnoticed by the writer) did not leave a good impression of the research team applying for funding by using a wrong abbreviation for the name of the agency.

The designers of AutoCorrect recognized this limitation of the adaptive part of AutoCorrect, and they added an adaptable component (see Figure 3) that allows users to overwrite existing modifications of the adaptive system. With this extension in place AutoCorrect replaces the original sentence of Figure 2 correctly.

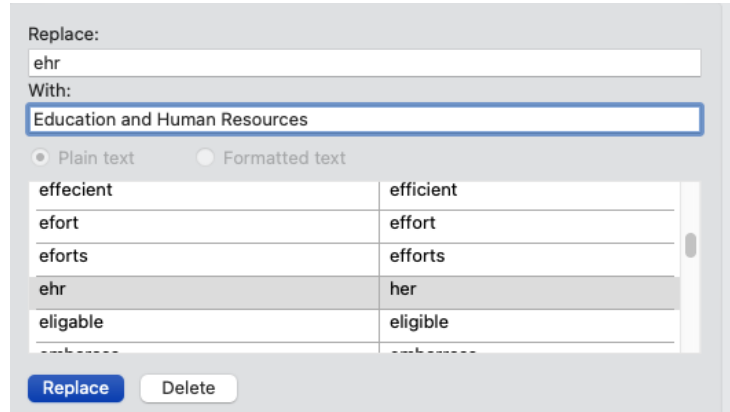


Fig. 3. The Interface of Auto-Correct to Support Adaptations by Users.

Beyond adaptive and adaptable components AutoCorrect includes a third heuristic: in cases where it identifies a potential misspelling with the adaptive component, but it is unsure which replacement should be chosen it returns control to the users with a set of possible modification and delegates the control to the user to decide (see Figure 4). This feature illustrates a simple heuristic for a symbiotic relationship between adaptive and adaptable features (further discussed later in the paper).

There are several additional features in AutoCorrect relevant for the theme of AI and/versus EUD including: (1) turning AutoCorrect off completely, and (2) ignoring the suggestions made by the adaptive systems.

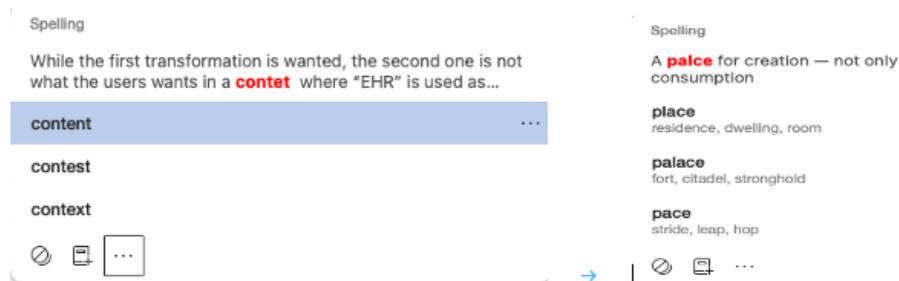


Fig. 4. Two Examples simplifying the adaptable component with suggestions from the adaptive component.

4 Adaptive Systems

Adaptive systems change their behavior automatically, driven by context-aware mechanisms [9] including models of their users and of specific tasks. Adaptive systems are important because the “typical” user of a system does not exist; there are many different users, and the requirements of an individual user usually change with experience [10]. Simple classification schemes based on stereotypes, such as novice,

intermediate, or expert users, are inadequate for complex knowledge-based systems because they do not take contexts into account. One of the central objectives of user modeling in Human-Computer Interaction (HCI) [11] is to address the problem that systems will be unable to interact with users cooperatively unless they have some means of finding out what individual users really know, do and intend. Techniques to achieve these objectives are: (1) being told by the users (e.g., by questionnaires, setting preferences, or specification components), and (2) being able to infer it from the user's actions (e.g., by analyzing usage data).

This section will describe some of the more prevalent benefits and drawbacks associated with the development and use of *adaptive* systems.

Benefit: Reducing Information Overload with Personalization. Personalization techniques are widely used in major technological developments; some prominent examples are:

- *Recommender systems* [12] assist customers in buying books or selecting movies of special interest for them. The basic idea of these systems is to present a selection of items to users which correspond closely to their specific interests. The collected data is based on items users have bought, recently viewed, and rated, and suggestions of interesting collaborations as identified by “big data” analyses [13];
- *Intelligent Tutoring Systems* [14] employ adaptive components by dynamically adjusting the level or type of course content based on an individual student's abilities contained in user models. They identify *zones of learnability* [15] by determining the gap between what a student knows and what a student is supposed to learn (e.g. according to defined learning goals) by moving students through a personalized learning path to prescribed learning outcomes and skill mastery.

Benefit: Becoming Aware of Unknown Things with Information Delivery. Information access (based on “pull” approaches) and information delivery (based on “push” approaches) are two approaches for obtaining information [1]. Information access relies on user-initiated searches, while information delivery is a system-initiated presentation of information that is intended to be relevant to the user's task. Information access schemes help the designer *articulate* information needs, while information delivery schemes *infer* information needs. Support for information access is indispensable since designers must have support to search for information when they perceive the need. Information delivery is a complementary approach that is particularly important when users are not motivated to look for information or are not aware of the existence of relevant information.

The fundamental challenge for information delivery systems is not to provide more information “to anyone, at any time, and from anywhere,” but to keep them quiet most of the time by exploiting context-aware mechanisms for identifying “*the ‘right’ information, at the ‘right’ time, in the ‘right’ place, in the ‘right’ way, to the ‘right’ person.*”[9].

Drawback: Personalization leading to a Lack of Shared Experiences and Common Understanding. Personalization [9] supported by adaptive system components addresses the fundamental issue that the scarce resource in the digital age is not information but human attention. Simon [16] illustrates this fact with convincing

examples that design representations suitable for a world in which the scarce factor is information may be exactly the wrong ones for a world in which the scarce factor is attention. Reducing the information is highly desirable but it represents a design trade-off [17] that needs to be taken into account, and the pros and cons need to be carefully evaluated. One of the things that binds a culture together is that people are exposed to the same relevant information presented from different points of view. One of the pitfalls of adaptive systems and user models is that they can and will create filter bubbles and echo chambers [18]. The polarization based on individual universes of information has become a defining characteristic in our world of today. Grounded in interests captured by big data, citizens will be grouped into types and be confronted with information that corresponds to their preferences and conforms to their beliefs which leads to *group think* [19]. There are numerous prominent examples such as (1) people in the USA watching CNN or Fox News will form beliefs and act accordingly as they would live in two different worlds and (2) the Corona virus has split societies around the world in totally opposed camps believing in vaccination or totally opposing it. To address global problems (such as Corona, climate change, digitalization, fake news, and the widening economic divide) requires the efforts of large communities and massive coordination to create a common understanding based on shared experience [20]. Groupthink, filter bubbles, and echo chambers represent serious obstacles for a future world in which people make conscious efforts to create common ground and avoid further polarization.

Drawback: Lack of Meaningful Explanations. Many current systems (specifically AI systems based on correlations within big data sets [13]) are unable to provide understandable explanations about their behavior. The inner workings of such systems are “black boxes”: they provide recommendations and answers, but they are unable to explain the underlying rationale needed for empowering users to adapt systems to their needs.

A simple example (encountered a couple of years ago by the author) occurred in the context of using the navigation system of a rental car. The fact that it was a rental car limited our familiarity with the functionality of the navigation system. We programmed the system to guide us the quickest way to our destination. Continuing our trip with the navigation system in operation, it advised us at every exit to leave the freeway. Without any way to query our navigation system and because we were familiar with the geography, we decided not to follow the advice that the system repeated at every exit. Unable to explore the differences between the system’s knowledge and ours for choosing the most preferable route, we followed our plan and shut the system off.

The navigation system acting as an adaptive system was unable to describe its intention in a way that we were able to understand. Drivers do not have any way to query these systems and they cannot adopt individual user’s perspectives to determine what statement would satisfy them. They cannot convey confidence in the route they have selected (other than giving a probabilistic estimate of the time differential for alternative routes), whereas we wanted them to explain the assumptions they are making for their recommendation.

When we returned the car and described the encountered problem, the rental car agent provided us with an explanation: somewhere deep down in a complex web of options there was a flag “do not use freeways” — and the previous renter had selected

this option as part of the adaptable features. An intelligent navigation system would have allowed us to ask the question “why should we leave the freeway” and then guided us to find the flag that was responsible for this behavior.

5 Adaptable Systems

New design methodologies [3, 21] explore that the once sharp distinction between users and developers of software is fading away, and many users are starting to take control of shaping software with adaptable systems to their needs by engaging in end-user development. Adaptable systems provide foundations for “democratizing innovation” as argued by von Hippel [7]: “*Users that innovate can develop exactly what they want, rather than relying on manufacturers to act as their (often very imperfect) agents. Moreover, individual users do not have to develop everything they need on their own: they can benefit from innovations developed and freely shared by others*”. Figure 3 illustrates the adaptable component of Auto-Correct to overwrite the changes made by the adaptive component. Another simple but useful adaptable system component supports the development of macros in MS-Word.

On a global scale, adaptable systems can be supported by: (1) offering task-specific languages supporting human problem-domain interaction with domain-oriented design environments [22]; (2) providing programming environments that protect users from low-level computational drudgery [23]; (3) supporting customization, reuse, and redesign effectively [24]; and (4) tailoring software applications at use time with component-based approaches [25];

This section will describe some of the benefits and drawbacks associated with the development and use of *adaptable* systems.

Benefit: Creating Support Environments for Adaptable Systems with Meta-Design. *Meta-design* (“design for designers”) [26] is a theoretical framework to conceptualize and to cope in unique ways with design problems. It is focused on open-ended co-design processes in which all the involved actors actively participate in different ways. It is grounded in the fundamental assumption that design is not a matter of getting rid of the emergent, but rather of including it and making it an opportunity for more creative and more adequate solutions to problems by supporting adaptable systems [27].

Research resulted in the following design requirements for encouraging end-users to engage in adaptable extensions [28]: (1) making changes must seem possible; (2) changes must be technically feasible; (3) benefits must be perceived by the stakeholders who do the work associated with the adaptations; (4) low barriers must exist for sharing changes; and (5) the original designers acting as meta-designers must be willing to share control of how systems will be used, which content will be contained, and which functionality will be supported.

Benefit: Putting the Problem Owners in Charge. A challenge for many software systems is the growing importance of application domain knowledge held by domain experts rather than by software developers, who suffer from a “thin spread of

application domain knowledge” [29]. Another challenge is the need for open, evolvable systems that can adjust to fluctuating and conflicting requirements.

A interview that we conducted with a geoscientist highlights the importance of these challenges that can be addressed by adaptable components. He uses several existing domain-specific software systems to analyze his research data. However, those systems cannot provide complete solutions to his problems as his research unfolds and his understanding of the problem, data, and results progresses. He said: *“I spend on average an hour every day developing software for myself to analyze the data I collected because there is not any available software. Even if there is a software developer sitting next to me, it would not be of much help because my needs vary as my research progresses and I cannot clearly explain what I want to do at any moment. Even if the software developer can manage to write a program for me, I will not know if he or she has done it right without looking at the code.”*

He continued *“so I spent three months to gain enough programming knowledge to get by. Software development has now become an essential task of my research, but I do not consider myself a software developer, and I don’t know many other things about software development.”*

This example provides evidence for democratizing innovation (see comment from von Hippel [7] earlier in the paper) and that software development is no longer the exclusive activity of professional software engineers. *Domain experts* being the owners of problems such as this geoscientist are engaged in intensive software development and adaptable systems provide important support environments to address this challenge. In education the Scratch project [23] very successfully made it easy for millions of children of all ages, backgrounds, and interests to engage in creating and remixing, and not just in browsing and interacting.

Drawback: Participation Overload. Adaptable systems open unique new opportunities for mass collaboration and social production [30], but these engagements are not without drawbacks. One such drawback is that humans may be forced to cope with the burden of being active contributors in personally irrelevant activities leading to a *participation overload* [31]. “Do-it-yourself” societies empower humans with powerful tools; however, they force them to perform many tasks themselves that were done previously by skilled domain workers serving as agents and intermediaries. Although this shift provides power, freedom, and control to users and customers, it also has urged people to act as contributors in contexts for which they lack the experience that professionals have at their disposal.

Drawback: Complexity and Heterogeneity of End Users. End users are a diverse group with varying levels of technical expertise. The needs and preferences of end users can vary widely depending on the context and domain, making it difficult to design EUD systems that meet everyone's needs.

Drawback: The Tension between Standardization and Improvisation. Adaptable systems create inherent tensions between standardization and improvisation. The SAP Info [32] argues to reduce the number of customer modifications: *“every customer modification implies costs because it has to be maintained by the customer. Each time a support package is imported there is a risk that the customer modification may have to be adjusted or re-implemented. To reduce the costs of such on-going*

maintenance of customer-specific changes, one of the key targets during an up-grade should be to return to the SAP standard wherever this is possible.” Finding the right balance between standardization (which can suppress innovation and creativity) and improvisation (which can lead to a Babel of different and incompatible versions) represents an important design trade-off (e.g., in open-source environments, in which forking leads developers in different directions).

Summary. Table 1 summarizes the major distinction between adaptive and adaptable systems.

Table 1. A Comparison and Differentiation between Adaptive and Adaptable Systems.

	Adaptive Systems	Adaptable Systems
Definition	suggestions generated by systems for specific tasks and users	users actively change the functionality of the system leading to distributed control
Knowledge	contained in the system; projected in different ways	knowledge is curated, modified, and extended by users resulting in living systems
Strengths / Benefits	little (or no) effort by users; no special user knowledge is required; work <i>for</i> people	users are in control; users know their tasks best; work <i>with</i> people
Weaknesses / Drawbacks	users lack control; shared understanding is reduced resulting in filter bubbles; lack of explainability; skill degradation	users must do substantial work (participation overload); require a learning effort; create a tool mastery burden; systems may become incompatible
Mechanisms required	models of users and tasks; context awareness; big data resources; intelligent agents	meta-design environments supporting modifiability, tailorability, and evolution
Application domains	recommender systems, intelligent tutoring systems, conversational agents	open systems, co-designed systems, end-user development
Primary Techniques	automation grounded in Artificial Intelligence (AI) approaches	human involvement grounded in Intelligence Augmentation (IA) approaches

6 Challenges for the Future

This section will discuss how the framework based on the analysis of adaptive and adaptable systems in the previous sections can provide design ideas and guidelines for differentiating and integrating AI and EUD and thereby contributing theoretical concepts and foundations for the field of end-user development.

6.1 Example: ChatGPT — the Newest “Miraculous” AI Development

As mentioned earlier *ChatGPT* (Chat Generative Pre-trained Transformer; <https://en.wikipedia.org/wiki/ChatGPT>) is credited with miraculous abilities to create unique and far-reaching new opportunities for Natural Language Understanding, Question Answering, Content Creation, and Education. It is a conversational AI program (a “chatbot”) developed by OpenAI and launched in November 2022. It has generated worldwide attention and the initial assessments combine *praise* such as

- a significant advancement that can produce articles in response to open-ended questions that are comparable to good high school essays;
- the shift to making search engines more conversational being able to confidently answer questions rather than offering links to other websites could change the advertising-fueled search business;
- changing our mind about how we work, how we think, and what human creativity really is;
- OpenAI is gathering data from ChatGPT to further train and fine-tune ChatGPT in the future;

with *fundamental criticisms* such as

- it poses risks if it does not get its facts right;
- its fluency is an illusion that stems from the combination of massive amounts of data, immense computing power, and novel processing techniques;
- it is not capable of understanding the context or meaning of the words it generates; it can only produce text based on the probabilities of certain words or sequences of words appearing together, based on the training data it has been given;
- it is difficult for humans to scrutinize the answer produced;
- it will give sometimes answers that are not related to the questions asked, and it will occasionally make up facts or “hallucinate” outputs.

Figure 5 provides an example showing ChatGPT’s response to a question relevant to the theme of this paper.

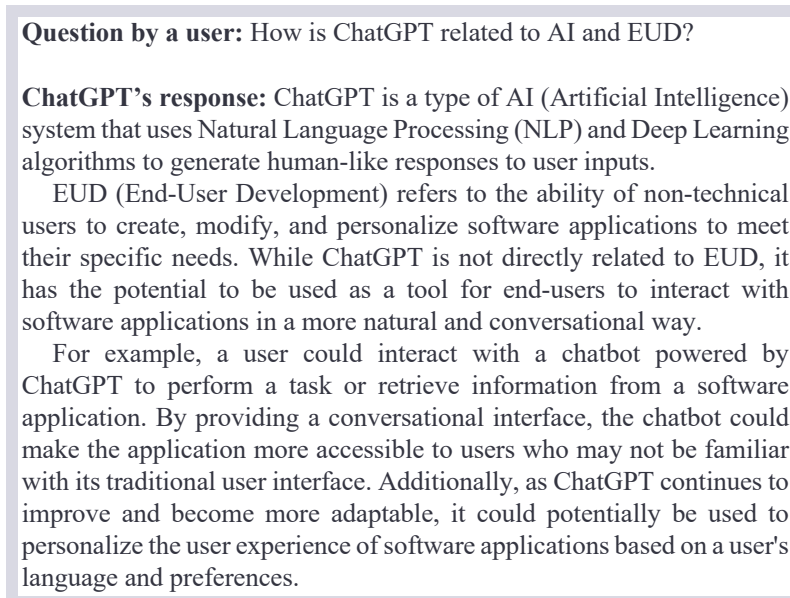


Fig. 5. ChatGPT in action

Figure 6 shows an architectural overview diagram of the two examples (AutoCorrect and ChatGPT) indicating adaptive AI components and adaptable EUD components.

The same analysis can be applied to understand adaptive and adaptable components of the current version of ChatGPT can provide theoretically grounded design ideas for its future development by exploring questions such as:

- Should further developments of ChatGPT strengthen the adaptive and/or adaptable components (in which way and to which extent)?
- Can users trust the answers provided by ChatGPT?
- Can ChatGPT support and enhance EUD capabilities when learners and workers use its answers not as final results but as additional commentaries and creative prompts enriching their further analysis?

A critical design challenge that has already generated numerous widely diverging views is to understand the implications of ChatGPT for *learning, teaching, and education* [33]. With ChatGPT being around only since late 2022, there are many unknowns about how it will be used, adopted, and further developed in the coming years. To increase our understanding much can be learnt from the controversial assumptions that governed the original discussion of Massive Open Online Courses (MOOCs) when they first appeared a decade ago [34]. Some of the controversial issues are:

- Will ChatGPT enhance or replace the necessity for writing essays?
- Where to draw the line between ChatGPT being used as an assistant for writing (the EUD approach) versus letting it take over the whole process (the AI approach)?

- Will its availability contribute to making the “flipped classroom” a more attainable reality because it makes relevant information readily available?
- Should teachers and learners become informed about the promises and pitfalls of ChatGPT and develop new and different skills?
- Which (current and future) mechanisms are required to turn ChatGPT from a black box into a glass box thereby empowering learners to scrutinize the answers provided?

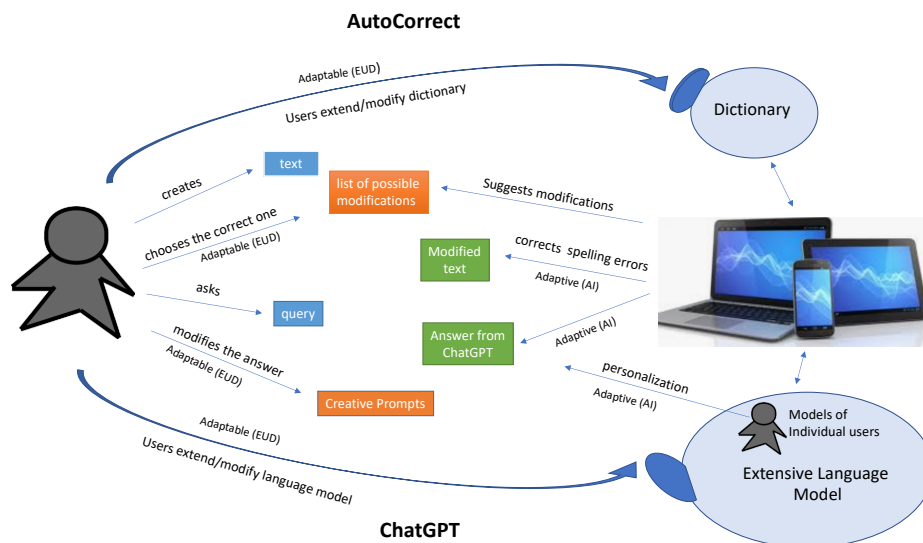


Fig. 6. The Intertwining of Adaptive and Adaptable components in AutoCorrect and ChatGPT <<the symbiotic options are colored in orange>>

6.2 Symbiosis between Adaptive and Adaptable Systems

In the previous sections of the paper and in Table 1, benefits and drawbacks of adaptive and adaptable systems were identified [35]. This analysis can serve as the foundation to create a *symbiosis* between adaptive and adaptable systems [36] and between AI and EUD approaches [37].

An adaptive system can provide valuable input to an adaptable system. As indicated in the AutoCorrect example, the adaptive component could generate a list of suggestions to support the adaptation process by the user to choose the correct choice. Answers generated by the adaptive component of ChatGPT (see Figure 5) maybe most valuable if not seen as “final”, but as starting points and inspirations for users to further adapt and improve the information content. In adaptive approaches, models of users and tasks can be inferred indirectly from interactions with systems whereas in adaptable approaches they can be described explicitly via specifications component [11].

Another rationale to support the integration is to compare ChatGPT and EUD with the framework of fast and slow thinking [38]. Fast thinking is automatic, intuitive, and effortless but prone to errors and biases. Slow thinking is controlled, rational, creative,

and more accurate and reliable. Kahneman's research demonstrates that human cognition is a *mixture* of these two modes of thinking. ChatGPT engages very successfully in "fast thinking" operations and can and should be complemented by the "slow thinking" component of EUD that verifies and transcends the information given [39].

Identifying and exploiting symbiotic relationships between adaptive and adaptable components can contribute to the objective to improve the *quality of life* in the digital age [6]. Digitalization will transform the world, but it does not mandate a single deterministic outcome and design trade-offs can spark efforts toward syntheses that lead to new levels of understanding and can productively exploit the best mixes between opposing choices. Automation (supported by adaptive components and AI approaches) can be a two-edged sword [40, 41]: (1) it is a *servant*, relieving humans of the tedium of low-level operations, and freeing them for higher cognitive functions or (2) it can reduce the status of humans to that of *button pushers*, and can strip their work of its meaning and satisfaction and eliminate learning opportunities.

The desirability of adaptable approaches and EUDs approaches to positively contribute to quality of life objectives is grounded in the fact that human beings value things and relationships for which they have to make an effort in obtaining them and in which they find purpose, enjoyment, and flow states [42]. Humans (in the context of *personally meaningful* tasks) enjoy 'doing' and 'deciding', they want control and autonomy, and they enjoy the process, and not just the final product [40]. But these objectives come with a price tag: they require time, engagement, and learning, and may lead (as indicated earlier) to *participation overload* [28, 31] which can be addressed and reduced by carefully designed adaptive components.

6.3 Design Guidelines

Adaptive and adaptable systems will be important components of future socio-technical environments. This section summarizes some *design guidelines* grounded in the frameworks and examples discussed in the paper.

- *Identify the design trade-offs associated with different approaches* including: to understand the benefits and pitfalls of adaptive and adaptable systems their uses must be situated and explored in specific contexts because there are no decontextualized sweet spots in design problems [5].
- *Strengthen the Benefits of Adaptive Systems* including: to reduce information overload, deliver unknown functionality and information, support personalization to focus attention, complement the tool approach with intelligent agents [43].
- *Strengthen the Benefits of Adaptable Systems* including: to create sociotechnical environments that empower domain experts to engage actively in the continuous development of systems rather than restricting them to using existing systems; support meta-design at design time for creating solution spaces in which users can create their own solutions to fit their needs at use time [28].

- *Be aware of the Drawbacks of Adaptive Systems* including being enclosed in filter bubbles and echo chambers; lack of explainability of the algorithmic decisions; and privacy intrusions [18].
- *Be aware of the Drawbacks of Adaptable Systems* including participation overload; incompatible version of systems; lack of rewards and recognition of contributions [5].
- *Give Humans Control over Technology* including: to identify the right mix between computer-based automation (replacing human beings) and human control (empowering human beings) [44].
- *Explore the Opportunities for Creating Symbiotic Relationships between Adaptive and Adaptable Systems* including adaptive systems analyze what exists and can provide foundations for how new functionality can be added with adaptable components; support user-controlled adaptation with system-generated adaptive suggestions [36].

6.4 Implication

Most of the currently existing AI approaches and visions are over-inclusive and reflect the common tendency to use the “AI” label as a catchall marketing phrase. AI is a “suitcase word” to which people attribute multiple meanings [45]. There is a need to differentiate AI and explore complementary approaches such as EUD.

This paper attempts to use the distinction between adaptive and adaptable approaches to show that AI can contribute not only by replacing human beings but in a synergistic fashion empowering them to act as active contributors in EUD to transcend “how things are” by exploring “*how things could or should be*” [16]. An essential challenge is that AI and EUD will find more new ways to communicate with each other [37]. Because the most fundamental problems that our world is facing are wicked and ill-defined [46], the future must remain something to which all stakeholders can make contributions. Providing all citizens in EUD approaches with the means to become co-creators of new ideas, knowledge, and products in personally meaningful activities presents one of the most exciting innovations and transformations with profound implications in the years to come.

7 Conclusions

To enrich rather than limit human lives with AI and EUD systems, discourses and investigations must not only be focused on technological issues but explore motivation, control, ownership, and autonomy. A better understanding is required of whether the technologies of the future (1) will provide us with more time, less stress, more control, and enhance human creativity or (2) will cause a shift in authority from humans to algorithms (especially with tools that humans do not understand and that cannot provide us with explanations about their actions). The relationship between AI and EUD and between adaptive and adaptable systems should not be conceptualized and driven

forward by competition, but by exploiting symbiotic relationships in human-centered design approaches.

While the growth of technologies such as AI and EUD will continue in the years to come, the inevitability of any particular future is not. To improve the *quality of life* in the digital age requires the co-design of social and technical systems and requires models and concepts that exploit the social context in which the systems will be used. The paper documents frameworks and illustrates with real-world examples identified by empirical research how to differentiate and integrate AI and EUD approaches with adaptive and adaptable systems components in socio-technical environments towards achieving these objectives.

8 Acknowledgments

The author wishes to thank Daniela Fogli, Anders Mørch, Antonio Piccinno, and the three reviewers who provided insightful comments that greatly improved earlier versions of this article.

9 References

1. Brown, J.S. and P. Duguid, *The Social Life of Information*. 2000, Boston, MA: Harvard Business School Press.
2. Brockman, J., *Possible Minds: Twenty-Five Ways of Looking at AI*. 2019: Penguin Press.
3. Paterno, F. and V. Wulf, eds. *New Perspectives in End User Development*,. 2017 Kluwer Publishers
4. Calvo, R.A. and D. Peters, *Positive Computing —Technology for Wellbeing and Human Potential*. 2014, Cambridge, MA: MIT Press.
5. Fischer, G., Design Trade-Offs for Quality of Life *ACM Interactions* 2018. 25 (1): p. 26-33.
6. Friedman, B. and D.G. Hendry, *Value Sensitive Design: Shaping Technology with Moral Imagination*. 2019, Cambridge, MA: MIT Press.
7. von Hippel, E., *Democratizing Innovation*. 2005, Cambridge, MA: MIT Press.
8. Burnett, M., What is End-User Software Engineering and Why Does It Matter?, in *End-User Development*, V. Pipek, et al., Editors. 2009, Springer: Heidelberg. p. 15-28.
9. Fischer, G., Context-Aware Systems: The ‘Right’ Information, at the ‘Right’ Time, in the ‘Right’ Place, in the ‘Right’ Way, to the ‘Right’ Person, in *Proceedings of the Conference on Advanced Visual Interfaces (AVI)*, G.T.e. al, Editor. 2012, ACM: Capri, Italy (May). p. 287-294.
10. Mackay, W.E., Triggers and Barriers to Customizing Software, in *Proceedings of CHI'91 Conference on Human Factors in Computing Systems*, S.P. Robertson, G.M. Olson, and J.S. Olson, Editors. 1991, ACM: New York. p. 153-160.
11. Fischer, G., User Modeling in Human-Computer Interaction. *User Modeling and User-Adapted Interaction (UMUAI)*, 2001. 11(1): p. 65-86.

12. Terveen, L. and W. Hill, Beyond Recommender Systems: Helping People Help Each Other, in *Human-Computer Interaction in the New Millennium*, J.M. Carroll, Editor. 2001, ACM Press: New York. p. 487-509.
13. Mayer-Schönberger, V. and K. Cukier, *Big Data*. 2013, New York, NY: Houghton Mifflin Harcourt.
14. Mousavinasab, E., et al., Intelligent tutoring systems: a systematic review of characteristics, applications, and evaluation methods. *Interactive Learning Environments*, 2021. 29(1): p. 142-163.
15. Vygotsky, L., *Thought and Language*. 1986, Cambridge, MA: MIT Press.
16. Simon, H.A., *The Sciences of the Artificial*. third ed. 1996, Cambridge, MA: The MIT Press.
17. Fogli, D., et al., Exploring design trade-offs for achieving social inclusion in multi-tiered design problems. *Behaviour & Information Technology*, 2020. 39(1): p. 27-46.
18. Pariser, E., *The Filter Bubble: How the New Personalized Web Is Changing What We Read and How We Think*. 2021: Penguin Books.
19. Janis, I., *Victims of Groupthink*. 1972, Boston: Houghton Mifflin.
20. Clark, H.H. and S.E. Brennan, Grounding in Communication, in *Perspectives on Socially Shared Cognition*, L.B. Resnick, J.M. Levine, and S.D. Teasley, Editors. 1991, American Psychological Association. p. 127-149.
21. Binder, T., et al., *Design Things*. 2011, Cambridge, MA: MIT Press.
22. Norman, D.A., *Things That Make Us Smart*. 1993, Reading, MA: Addison-Wesley Publishing Company. 290.
23. Resnick, M., et al., Scratch: Programming for All. *Communications of the ACM*, 2009. 52(11): p. 60-67.
24. Morch, A., Three Levels of End-User Tailoring: Customization, Integration, and Extension, in *Computers and Design in Context*, M. Kyng and L. Mathiassen, Editors. 1997, MIT Press: Cambridge, MA. p. 51-76.
25. Wulf, V., V. Pipek, and M. Won, Component-based tailorability: Enabling highly flexible software applications. *International Journal of Human-Computer Studies*, 2008. 66: p. 1-22.
26. Costabile, M.F., et al., Advancing End User Development Through Metadesign, in *End User Computing Challenges and Technologies: Emerging Tools and Applications*, S. Clarke, Editor. 2008, IGI Publishing. p. 143-167.
27. National-Research-Council, *Beyond Productivity: Information Technology, Innovation, and Creativity*. 2003, Washington, DC: National Academy Press.
28. Fischer, G., D. Fogli, and A. Piccinno, Revisiting and Broadening the Meta-Design Framework for End-User Development, in *New Perspectives in End User Development*, F. Paterno and V. Wulf, Editors. 2017, Kluwer Publishers: Dordrecht, The Netherlands. p. 61-97.
29. Curtis, B., H. Krasner, and N. Iscoe, A Field Study of the Software Design Process for Large Systems. *Communications of the ACM*, 1988. 31(11): p. 1268-1287.
30. Benkler, Y., *The Wealth of Networks: How Social Production Transforms Markets and Freedom*. 2006, New Haven: Yale University Press.
31. Jenkins, H., *Confronting the Challenges of Participatory Cultures: Media Education for the 21st Century*. 2009, Cambridge, MA: MIT Press.

32. Fischer, G., K. Nakakoji, and Y. Ye, Meta-Design: Guidelines for Supporting Domain Experts in Software Development. IEEE Software 2009. September/October: p. 37-44
33. Collins, A., What's Worth Teaching: Rethinking Curriculum in the Age of Technology. 2017, New York: Teachers College Press.
34. DeCorte, E., L. Engwall, and U. Teichler, eds. From Books to MOOCs? Emerging Models of Learning and Teaching in Higher Education. 2016, Portland Press (Wenner-Gren International Series Volume 88): London.
35. Maes, P. and B. Shneiderman, Direct Manipulation vs. Interface Agents. Interactions, ACM Press, 1997. 4(6): p. 42-61.
36. Oppermann, R., Adaptively Supported Adaptability. International Journal of Human-Computer Studies, 1994. 40(3): p. 455-472.
37. Markoff, J., Machines of Loving Grace (The Quest for Common Ground Between Humans and Robots). 2016: Harpercollins
38. Kahneman, D., Thinking, Fast and Slow. 2011, New York: Farrar, Straus and Giroux.
39. Bubeck, S. and V. Chandrasekaran, et al. . Sparks of artificial general intelligence: Early experiments with GPT-4. 2023; Available from: <https://arxiv.org/pdf/2303.12712.pdf>.
40. Shneiderman, B., Human-Centered AI 2022: Oxford University Press.
41. Carr, N., The Glass Cage — Automation and US. 2014, New York: Norton.
42. Csikszentmihalyi, M., Creativity — Flow and the Psychology of Discovery and Invention. 1996, New York, NY: HarperCollins Publishers.
43. Fischer, G., Domain-Oriented Design Environments. Automated Software Engineering, 1994. 1(2): p. 177-203.
44. Fischer, G. and K. Nakakoji, Beyond the Macho Approach of Artificial Intelligence: Empower Human Designers - Do Not Replace Them. Knowledge-Based Systems Journal, Special Issue on AI in Design, 1992. 5(1): p. 15-30.
45. Minsky, M., The Emotion Machine. 2007: Simon & Schuster.
46. Rittel, H. and M.M. Webber, Planning Problems are Wicked Problems, in Developments in Design Methodology, N. Cross, Editor. 1984, John Wiley & Sons: New York. p. 135-144.