Proficiency-Aware Systems: Designing for User Reflection in Context-Aware Systems

Jakob Karolus, Paweł W. Woźniak

Abstract:
In an increasingly digital world, intelligent systems support us in accomplishing many everyday tasks. With the proliferation of affordable sensing devices, inferring user states from collected physiological data paves the way to tailor-made adaptation. While estimating a user's abilities is technically possible, such proficiency assessments are rarely employed to benefit the user's task reflection. In our work, we investigate how to model and design for proficiency estimation as part of context-aware systems. In this paper, we present the definition and conceptual architecture of proficiency-aware systems. The concept is not only applicable to current adaptive systems but provides a stepping stone for systems which actively aid in developing user proficiency during interaction.

ACM CCS: Human-centered computing → Human computer interaction (HCI) → Interaction paradigms

Keywords: Proficiency, adaptive systems, proficiency-aware systems, interaction paradigm.

1 Introduction

We spend an increasing fraction of the time in our lives using computers. Consequently, we develop skills at using interfaces that we know. Despite that, we often encounter barriers in starting to use a new system. Imagine you are traveling in a foreign country. You arrive at the airport and want to use an information kiosk to find your way to the car rental. You approach the kiosk only to discover that the interface is displayed in a language that you do not understand. You cannot even identify where the language menu is as the controls are displayed in an alphabet that you do not know. Eventually, you are unable to use the system. If the kiosk were able to sense that you do not know the interface language, it could offer you relevant assistance.

As we rely on digital systems for a variety of tasks in our everyday lives, we tolerate an increasing dependence on such intelligent systems. Yet, background and knowledge vary substantially among users, and simple one-size-fits-it-all approaches often fail to tailor to the needs of specific user groups. Recognition and subsequent adaptation to user needs and skills are essential to building meaningful future interactive technologies. Human-computer interaction (HCI) research has historically made use of user modeling [9] and context-awareness [21] to cater to individual user characteristics. One major driver of this development is the proliferation of affordable sensing devices and technology to quantify "the context in which [systems] are run" [21].

In this work, we take a closer look at the aspect of a user's proficiency and how it can inform future context-aware systems. We conceptualize the idea of *proficiency-aware systems*—a class of context-aware systems that are aware of a user's proficiency and adapt interaction accordingly. We postulate that a user's skill level is an integral part of the context of use, one to be considered on par with other elements of contexts, e.g., location or time.

Consequently, we provide a definition for proficiency-aware systems and the concept of system-specific proficiency and showcase how this concept fits into the existing concept of context-aware systems. While the latter are able to assess and rate users' abilities in a wide range of domains and applications, such proficiency assessments are rarely employed to benefit the user's reflection about a given task. It remains a challenge to understand how systems can adapt primarily for proficiency and provide means of reflection on proficiency. As we use more and more intelligent systems, we need to learn how to use them faster and more effectively.

To illustrate this challenge, we present the concept of proficiency in the domain of context-aware systems and discuss an accompanying conceptual architecture for building such systems. To facilitate future developments, we demonstrate classes (types) of proficiency-aware systems with examples from past research work. We conclude with an outlook on the opportunities and challenges for future proficiency-aware systems.

2 Proficiency-Aware Systems

Technological advances have developed the original definition of context-awareness proposed by Schilit in 1994 [21] which was updated over time [22, 6]. In Dey's definition "any information that can be used to characterize the situation of an entity" [6] should be considered by a context-aware system. Consequently, a context-aware system should only provide "relevant information" [6]. Importantly, "relevancy depends on the user's task" [6].

2.1 Towards Proficiency-Aware Systems

We argue that features of context-aware applications (cf. [6]) — while making use of context — are rarely employed for the user's benefit besides supporting task completion. We envision that such systems should be able to provide the means for users to understand and, if necessary, become aware of their own context. We postulate that a user's skill and knowledge are promising aspects of context to investigate in this regard [18].

Especially when it comes to understanding one's own lack of competence, humans tend to overestimate themselves [15]. Thus, most users might not appreciate a system judging their proficiency and hence do not develop an understanding of their own proficiency. Consequently, spending resources on improving one's skill is deemed wasteful and may only elicit a least-effort response from the user. This can potentially be dangerous as users can exploit assistance from systems jeopardizing proficiency development.

Ultimately, a lack of information about one's own proficiency leads to "satisficing" [11], where users cannot perform up to expectations. To conclude, proficiency is an underexplored aspect of context in context-aware systems. Understanding how we can leverage proficiency to assist users in their primary tasks and support users in understanding their own proficiency is a crucial aspect of proficiency-aware systems.

2.2 A Definition of Proficiency-Aware Systems

In this work, we present the concept of proficiency-aware systems, which describes a subset of context-aware systems that adapt to a user's proficiency. Proficiency is a multifaceted construct incorporating, among others, the user's inherent abilities, experience, and acquired skill.

The Oxford English Dictionary[1] defines proficiency as:

> **a.** The quality or fact of being proficient; the degree of competence attained; adeptness, skill *in* a particular field.
> **b.** A skill, a talent; (now frequently) a certain standard of skill acquired after a period of education or training.

Proficiency in Proficiency-Aware Systems

We derive our definition accordingly and specifically define proficiency as a user's skill *in* associated task domains. Those skills might be acquired, inherently present, or improved through experience. Note that proficiency is bound to "a particular field" of application. We represent this fact in our definition by allowing proficiency to be an aggregation of skills in associated task domains. By doing so, it is easier to operationalize and quantify proficiency, as specific tasks can be broken down, and the associated skill level can be obtained with ease. For example, it might be challenging to create a test to ascertain a user's proficiency in "write an essay about American history in plain English". However, it is feasible to combine their skill in "write in plain English" and the user's competence in American history.

For proficiency-aware systems, we define proficiency as:

> **Proficiency** is the aggregated construct of any skills, knowledge, competence, or experience of a person relevant to the interaction between the person and a system (the task domain).

Accordingly, we define a proficiency-aware system as:

> **Proficiency-Aware Systems** are systems that use estimates of a user's proficiency to tailor the interaction in their associated task domain(s). A **proficiency-aware system** is able to adapt content, presentation, and interaction accordingly to (1) support the user in accomplishing their task or (2) facilitate an understanding for the user of their own (lack of) proficiency.

Noteworthy, this definition allows us to draw parallels to context and context-aware systems as detailed above. At the core, a proficiency-aware system is simply a context-aware system that focuses on the user's proficiency. Nevertheless, the definition allows for more elaborate systems to support the users to reflect on their work — a vital design goal in HCI [2] — facilitating an understanding of their own proficiency.

Proficiency as an Aggregated Construct

In our conceptualization of proficiency-aware systems, proficiency does not equal task performance, i.e., it is not alpha power of brain waves or typing speed. Proficiency is an aggregated construct of relevant user skill, knowledge, and competence that may be inferred from measured metrics. For example, typing speed can serve as an indicator for a user's skill in typing on a keyboard.

[1] https://www.oed.com/

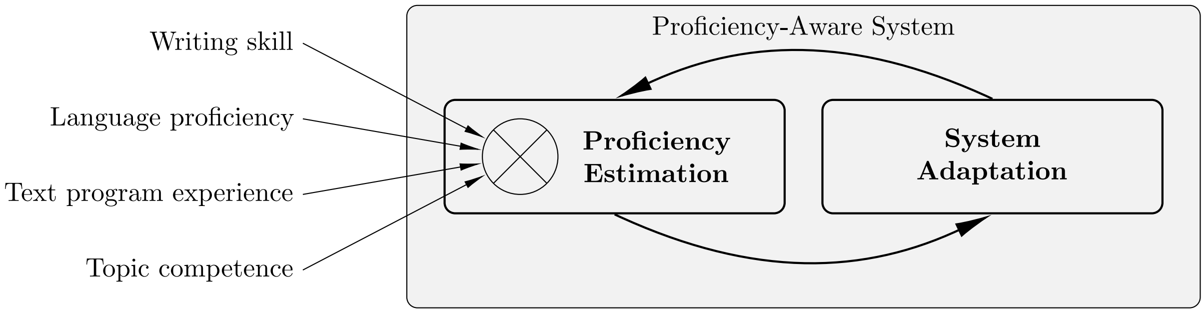

Figure 1: An example model of the states of a proficiency-aware system for text generation. Relevant proficiency is inferred by a set of specific factors (left side) and supplied to the system (generic depiction). The system itself does not necessarily have to be responsible for estimating individual factors contributing to a user's proficiency.

The aggregation of relevant user skills contributes to an understanding of the user's proficiency in the actual task at hand. The dynamic connection between quantifying a user's proficiency and adequately adapting to it is illustrated in Figure 1. It illustrates the idea of proficiency detection as a high-level concept, illustrating (possibly) relevant skills for a writing task.

While overall proficiency is usually difficult to measure accurately, individual contributing skill levels may be easier to estimate. In practice, it is often sufficient to consider only relevant skills that might influence a person's proficiency for the given task. Identifying these is a key design task when building a proficiency-aware system.

We illustrate this with an example: a professional editor for a newspaper is given the task to proofread an article about a breakthrough discovery in computer science. In this case, we can assume a high skill level in writing, a skill inherent to this particular user. Likewise, unless the editor has a computer science background, their competence about the topic is limited. This means that the editor is already at a disadvantage, as their limited knowledge about the topic impacts the proofreading task. Here, a proficiency-aware system might offer explanations for unfamiliar terms or even suggest articles to study as preparation for the task.

Further, let us consider two different contexts: a typical workday around 11 am and a stressful day just before the end of work. The editor's proofreading skill is unchanged, yet, the editor will have a harder time with the task within the second context. This is a typical example of a context-aware system and situational impairments [24]. Notice that we explicitly did not include short-term impairment factors in our definition of proficiency. Such factors should be modeled through context. Hence, a proficiency-aware system can make use of the context by deciding to support the editor for the second context, focusing on task completion. In contrast, if the context allows, the proficiency-aware system points out ways to increase the editor's knowledge about computer science.

For some systems, this set of skills is readily available and straightforward to measure. However, it is often impossible to quantify the skill level to a degree necessary to provide effective adaptation. Consider this next example: you are tasked with writing a proposal for a broad audience on how to make your city greener. This task requires skills in several areas, such as writing in plain and understandable English, structuring one's ideas concisely, or providing convincing arguments. The latter two skills are inherently difficult to estimate. Yet, it may not be necessary to model the complete proficiency to support the user. Here, measuring the skill of being able to write in understandable English can already be sufficient. This puts the writer in a better position to write the required proposal, not only for this specific topic but also for other topics.

2.3 Adaptive Systems and Intelligent Tutoring Systems

As we classified proficiency-aware systems as a subclass of context-aware systems, the concept can be applied to other adaptive systems. Here, we want to provide a closer scope for proficiency-aware systems to identify similarities and significant differences.

Proficiency is closely related to a person's inherent cognitive and physical abilities. Note that we deliberately did not include the impact of abilities in our definition. For example, proficiency in reading a book is naturally inhibited by reading disorders. While there exists work that investigates "ability-based optimization" [19, 20], long-term or permanent inhibitors of proficiency should be considered separately and appropriately modeled. The main strengths of proficiency-aware systems lie in the adaptation during runtime. Consequently, proficiency-aware systems are designated as adaptive systems, not adaptable systems.

Intelligent tutoring systems (ITS) show several similarities of proficiency-aware systems but are more closely related to user modeling approaches [9], including a student model and pedagogical module [4]. Here, a proficiency-aware system commonly only has access to the task environment (the context) and domain knowledge (the task domain) [4]. Consequently, in-depth user modeling is neither intended nor desired. However, elements of ITS that foster reflection for the user can provide valid adaptations for proficiency-aware systems.

2.4 Classes of Proficiency-Aware Systems

To give a structured overview of how recent research projects are related to the concept of proficiency-aware systems and show opportunities for new systems, we introduce a set of dimensions that categorize a proficiency-aware system, which we call classes. We note that this overview is not exhaustive and only highlight particular dimensions that are interesting for proficiency. Categorizations of context-aware systems are still applicable to proficiency-aware systems.

Interaction Duration and Frequency

A proficiency-aware system might be used once for a short duration (cf. the airport example) or be employed for a longer duration, such as a student learning system [7]. The first group relies on local adaptation based on ad-hoc context, thus preserving privacy if data is not shared beyond the interaction. In contrast, the latter group is able to provide a much more customized user experience, as systems potentially have access to an extensive history of data. However, the interaction duration of a proficiency-aware system is often predetermined by its usage scenario.

Adaptation Frequency

We distinguish between a dynamic proficiency-aware system that reacts to proficiency changes, adapting immediately, and a static one, allowing for periodic changes, e.g., at regular intervals. Thus, a dynamic proficiency-aware system directly reacts to a change in proficiency, handing control over to the proficiency estimation part. A static version leaves control at the adaptation side, only querying for an update of the proficiency estimation when required or scheduled.

Level of User Control

When interacting with a system, users perceive a sense of control when they feel that actions of the system are evoked by them. Similarly, a proficiency-aware system can present its users with a choice of adaptations by providing a recommendation, giving full control to the user. However, this method requires explicit actions by the user. In contrast, implicit adaptation allows for a seamless user experience while potentially leaving the user with a feeling of a loss of control.

Level of Communication

While a simple adaptive system might not need to explicitly communicate adaptation decisions to the user, a more elaborate proficiency-aware system may use this opportunity to provide adequate task feedback and means for improvements to the user as detailed in our definition (cf. Section 2). Consequently, we distinguish four types of feedback that a proficiency-aware system can offer: no feedback, simple performance-based feedback (e.g., scoring), raw measured data, and informed feedback. The latter two are specific for systems that employ physiological sensing. While raw data feedback is most straightforward, it might be challenging to interpret for laymen. On the other hand, informed feedback presents measured data in a way that is more understandable for the user. This approach, however, often requires extensive domain knowledge of the task, such as improving and detecting erroneous execution.

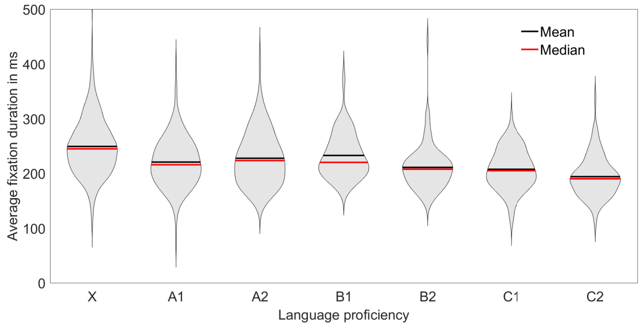

Figure 2: Average fixation duration given levels of language proficiency.

3 Understanding Existing Proficiency-Aware Systems

In this section, we look at selected research projects, which employ physiological computing for context-awareness. We classify these according to our definition of proficiency-aware systems (cf. Section 2) and illustrate the operationalization of proficiency that these systems use. Table 1 provides an overview of the categorization of highlighted research projects and mentioned examples in this paper into proficiency-aware system classes.

3.1 Short-Term, Static, User Control

Analyzing users' gaze properties is a major research area when it comes to predicting a user's skill and expertise in vision-dominant tasks. From simple reading assessments [17], through language proficiency [16] to distinguishing novices from experts [10]. Here, we discuss a recent example of a system employing robust gaze features for language proficiency awareness by Karolus et al. [14]. The system represents a proficiency-aware system with short-term, one-time-only user interaction. It exploits robust features such as average fixation duration to deliver a rough estimation — proficient enough to read or not. Figure 2 illustrates the relation between average fixation duration and associated language proficiency (CEFR[2]).

The system is able to judge whether a user can read a given sentence and suggests an appropriate adaptation. A possible interaction can be seen in Figure 3. However, the system does not yet provide a complete cycle as

[2] Common European Framework of Reference for Languages [5].

Table 1: Categorization of past research project examples and illustrative examples in this paper into proficiency-aware system classes. The diversity of the systems illustrates the complexity of design decisions involved in building a proficiency-aware system.

	Interaction Duration and Frequency	Adaptation Frequency	Level of User Control	Level of Communication
Research projects				
Robust gaze features [12]	short-term	static	user	none
BACh [25]	long-term	dynamic (cont.)	system	performance
Biocyb. loop [8], Player balancing [8]	long-term	dynamic (cont.)	system	none
EMGuitar [13]	long-term	dynamic (regular)	user/system	performance
Examples				
Airport information kiosk	short-term	static	user	none/informed
Green city proposal	short-/long-term	static/dynamic	user	informed
Newspaper editor	long-term	dynamic	user/system	perf./informed

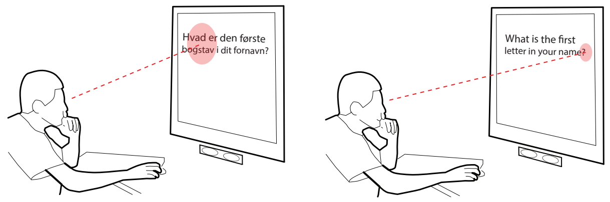

Figure 3: A possible adaptation for a gaze-based language-aware system.

defined in Section 2. This work illustrates one major challenge for adapting to proficiency: if the proficiency domain is known, it is possible to provide a possible alternative to display to the user. Yet, knowing that one is not proficient in a given language is not enough to suggest another display language. Here, a static approach to adaptation that gives the user full control over the adaption is most suited, for example, a language selection menu.

Consequently, static adaptation and yielding complete control to the user are most beneficial in scenarios where choosing a wrong adaption can potentially be worse than the current presentation.

3.2 Long-Term, Dynamic

For precise proficiency assessment, it is often beneficial to collect task-specific metrics. This allows for sufficient observation time or even continuous observation. In the following, we highlight two works that fall into this category but employ different forms of user control.

System Control Over Proficiency

If the system has full control over the adaptation, users may perceive a loss of agency. Here, adaptation frequency and noticeability of changes are crucial [3]. Similarly, this paradigm can be employed in physiological computing. BACh [25] is a system by Yuksel et al. that automatically adjusts the playing difficulty of piano pieces based on the user's cognitive workload. The system leverages fNIRS[3] to measure a player's current workload and switches to a higher difficulty if it falls below a defined threshold. Through BACh, users are able to learn pieces faster and more accurately than in the control condition. Here, the system relies on continuous observations, and when the acquired proficiency for a specific difficulty level is reached, it increases the difficulty to challenge the user. Players reported more efficient learning with BACh, although they had to relinquish control over how they approach learning the piece.

Research by Ewing et al. [8] takes this to the extreme by implementing a biocybernetic loop that feeds EEG measures directly to the system allowing continuous adaption of the difficulty in a Tetris game. In this case, the user receives no feedback on their proficiency, though a learning effect might still be achieved.

Consequently, it can be beneficial to restrict user control if the task can be divided into ordered levels of required proficiency. This is the case if the associated proficiency function is convex, making it possible to determine the gradient.

User Control Over Proficiency

Similarly, EMGuitar [13] is a guitar tutoring system that evaluates a user's playing proficiency for a given piece. The system uses Electromyography (EMG) to record muscle activity and classifies played guitar chords accordingly. In contrast to BACh, EMGuitar only suggests a new difficulty level in the form of tempo adjustments. The user is free to adhere to this recommendation or choose their own tempo. Players reported that the suggested tempi provided a good challenge and that they trusted in the system's assessment. This showcases that especially beginner and amateur players (as reported in [25, 13]) appreciate a *helping hand* in curating their learning process.

[3] Functional near-infrared spectroscopy. Measures levels of activation in the brain due to the hemodynamic response.

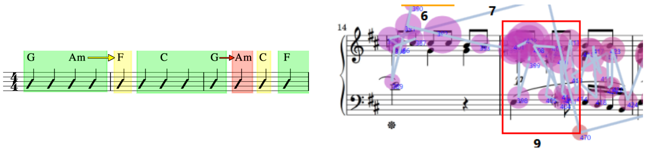

Figure 4: Domain-specific feedback for music tutoring systems. Guitar chords on the left, piano (with gaze overlay) on the right.

3.3 Communicating Proficiency

All three projects presented above lack an explicit feedback system for the user about their inferred proficiency. It is not apparent to the user how they can improve their skill at the given task. Both BACh and EMGuitar use global performance metrics to infer proficiency, such as task accuracy and user workload. Thus, the system contains information about when and how to adapt, but it does not contain data on the nature of the mistakes committed by the user. Consequently, the system cannot provide tailored feedback that goes beyond adjusting generic task difficulty. Thus, the burden of skill development is placed solely on the user.

On the one hand, simple feedback methods allow the systems above to detect a multitude of task errors and user flaws without detailed access to underlying task semantics. On the other hand, informed feedback tailored to the task domain can provide a more efficient learning experience. The examples presented show that balancing feedback is key for designing proficiency-aware systems. Figure 4 illustrates two examples of possible feedback for music tutoring applications.

4 Building Future Proficiency-Aware Systems

In the previous section, we have showcased several works that can be classified into proficiency-aware systems. Further, we argue that explicitly designing for proficiency-awareness opens new opportunities and challenges, which we highlight in the following.

There Is More To Proficiency Than Task Performance

Most context-aware systems rely on task performance for adaptation. This method is robust and often yields satisfactory results. However, this can only be successful if task performance is a good proxy for proficiency. We envision that future systems not only target immediate performance but strive to provide meaningful proficiency feedback and respective adaptation for the user. By doing so, users can reflect on their own proficiency [23] and are motivated to spend that extra step on improving. This is especially important if multiple factors influence task performance (cf. the proposal example). Consequently, **proficiency-aware systems should inherently motivate the user to improve their own skill levels**.

Communicating Proficiency Is Challenging

It remains a challenge of how to properly convey motivation and avoid judgement. For users, it might not be evident that the provided feedback is task-oriented, reducing its effectiveness. This directly relates to the level of control that the system offers to the user. If a system's decisions contradict a user's self-assessed proficiency, the user experience will be poor, and users will likely abandon using the system. Consequently, **transparency of communicated proficiency should be carefully considered** to gain the user's trust in the proficiency estimation.

Data Transparency and Availability Are Essential

Like most intelligent systems, proficiency-aware systems suffer from a lack of transparency during the decision-making process. Automated analysis of complex signals is often challenging to understand for laymen. Hence, users view those decisions with skepticism, but even developers and engineers find sensing modalities too complex. With the rise of explainable AI [1], the black-box nature of systems is not contemporary anymore. It should be evident to the user what and why data is collected and, most importantly, for what it is being used. We envision that the increasing availability of toolkits will help to remedy this issue. The more users and developers understand the characteristics of signals, the more likely they are to accept proficiency estimations based on sensed signals. Consequently, future **proficiency-aware systems should rely on established sensing modalities** if possible. If this is not an option, **transparency should be a guiding design principle**.

Non-Convex Proficiency Functions Can Often Be Simplified

Theoretically, adaptation is trivial if the system wants to guide the user towards a proficiency optimum. This, however, is only possible if the aggregation of proficiency is a convex function. For most problems, there exists at least a convex subspace for a given proficiency function (cf. guitar and piano tutor systems). Here, the system can guide the user towards a proficiency optimum effectively. Yet, it can be impossible to gauge an adequate proficiency gradient in some domains, likely because no universally agreed-on optimum proficiency exists or proficiency is highly subjective. This is particularly important in creativity domains, such as the arts. Improper assistance by the system could even result in a loss of quality. We argue that for most proficiency-aware systems, **a convex subspace exists, and it effectively guides the user towards improving their proficiency**. Recalling the arts example: a proficiency-aware system might identify the stroke fidelity of a painter, guiding the user in extending their armamentarium, but the system is unlikely to help them become a more creative painter through adaptation.

Learning Versus Proficiency Augmentation

Proficiency-aware systems can provide a temporary boost in proficiency, such as assisting a user with a given task. Proficiency augmentation is beneficial for temporary adaptation but may not result in lasting skill improvement. Rather, **a proficiency-aware system can be designed with the goal of supporting the user in permanently improving their level in task-associated skills**. By doing so, systems provide the opportunity for users to develop their skills, potentially having a lasting impact on proficiency. It remains a challenge to address and invoke a proper learning process in a proficiency-aware system. Domain-specific methods are most effective, yet they often conflict with the generalizability and universality necessary for adaptive systems.

5 Conclusion

Most of the techniques, associated methods, and approaches that we connected to proficiency-aware systems have been a focus of research for a long time. In this paper, we propose a conceptual architecture to structure our understanding of these systems, which builds on established paradigms such as context-awareness and physiological computing. We illustrate this concept by relating it to past research projects. Based on current shortcomings, we identify future challenges and opportunities for proficiency-aware systems. We believe that a shared perspective is essential to advance the understanding of requirements for the future of physiological computing. We envision our concept of proficiency awareness leading to new interaction paradigms, especially in the domain of ubiquitous computing.

Literature

[1] Adadi, A., and Berrada, M. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access 6* (2018), 52138–52160.

[2] Baumer, E. P., Khovanskaya, V., Matthews, M., Reynolds, L., Schwanda Sosik, V., and Gay, G. Reviewing reflection: On the use of reflection in interactive system design. In *Proceedings of the 2014 Conference on Designing Interactive Systems* (New York, NY, USA, June 2014), DIS '14, Association for Computing Machinery, pp. 93–102.

[3] Cechanowicz, J. E., Gutwin, C., Bateman, S., Mandryk, R., and Stavness, I. Improving player balancing in racing games. In *Proceedings of the First ACM SIGCHI Annual Symposium on Computer-Human Interaction in Play* (New York, NY, USA, Oct. 2014), CHI PLAY '14, Association for Computing Machinery, pp. 47–56.

[4] Corbett, A. T., Koedinger, K. R., and Anderson, J. R. Chapter 37 - Intelligent Tutoring Systems. In *Handbook of Human-Computer Interaction (Second Edition)*, M. G. Helander, T. K. Landauer, and P. V. Prabhu, Eds. North-Holland, Amsterdam, Jan. 1997, pp. 849–874.

[5] Council of Europe. *Common European Framework of Reference for Languages: Learning, Teaching, Assessment.* Applied Linguistics Non Series. Cambridge University Press, 2001.

[6] Dey, A. K. Understanding and Using Context. *Personal and ubiquitous computing* (2001), 4.

[7] D'Mello, S. K., Olney, A., Williams, C., and Hays, P. Gaze tutor: A gaze-reactive intelligent tutoring system. *Int. J. Hum. Comput. Stud. 70* (2012), 377–398.

[8] Ewing, K. C., Fairclough, S. H., and Gilleade, K. Evaluation of an Adaptive Game that Uses EEG Measures Validated during the Design Process as Inputs to a Biocybernetic Loop. *Front Hum Neurosci 10* (May 2016).

[9] Fischer, G. User Modeling in Human–Computer Interaction. *User Modeling and User-Adapted Interaction 11*, 1 (Mar. 2001), 65–86.

[10] Grindinger, T., Duchowski, A. T., and Sawyer, M. Group-wise similarity and classification of aggregate scanpaths. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications* (New York, NY, USA, Mar. 2010), ETRA '10, Association for Computing Machinery, pp. 101–104.

[11] Herbert A. Simon. Rational choice and the structure of the environment. *Psychological Review*, 63(2) (1956).

[12] Karolus, J., and Schmidt, A. Proficiency-Aware Systems: Adapting to the User's Skills and Expertise. In *Proceedings of the 7th ACM International Symposium on Pervasive Displays* (Munich Germany, June 2018), ACM, pp. 1–2.

[13] Karolus, J., Schuff, H., Kosch, T., Wozniak, P. W., and Schmidt, A. EMGuitar: Assisting Guitar Playing with Electromyography. In *Proceedings of the 2018 Designing Interactive Systems Conference* (Hong Kong, China, June 2018), DIS '18, Association for Computing Machinery, pp. 651–655.

[14] Karolus, J., Wozniak, P. W., Chuang, L. L., and Schmidt, A. Robust Gaze Features for Enabling Language Proficiency Awareness. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2017), CHI '17, ACM, pp. 2998–3010.

[15] Kruger, J., and Dunning, D. Unskilled and Unaware of It: How Difficulties in Recognizing One's Own Incompetence Lead to Inflated Self-Assessments. *Journal of Personality and Social Psychology 77* (1999), 1121–1134.

[16] Martínez-Gómez, P., and Aizawa, A. Recognition of understanding level and language skill using measurements of reading behavior. In *Proceedings of the 19th International Conference on Intelligent User Interfaces* (New York, NY, USA, Feb. 2014), IUI '14, Association for Computing Machinery, pp. 95–104.

[17] Rayner, K. Eye movements in reading and information processing: 20 years of research. *Psychol Bull 124*, 3 (Nov. 1998), 372–422.

[18] Rogers, Y. Moving on from Weiser's Vision of Calm Computing: Engaging UbiComp Experiences. In *UbiComp 2006: Ubiquitous Computing* (Berlin, Heidelberg, 2006), P. Dourish and A. Friday, Eds., Lecture Notes in Computer Science, Springer, pp. 404–421.

[19] Sarcar, S., Jokinen, J. P. P., Oulasvirta, A., Wang, Z., Silpasuwanchai, C., and Ren, X. Ability-Based Optimization of Touchscreen Interactions. *IEEE Pervasive Computing 17*, 1 (Jan. 2018), 15–26.

[20] Sarcar, S., Joklnen, J., Oulasvirta, A., Silpasuwanchai, C., Wang, Z., and Ren, X. Towards Ability-Based Optimization for Aging Users. In *Proceedings of the International Symposium on Interactive Technology and Ageing Populations* (New York, NY, USA, Oct. 2016), ITAP '16, Association for Computing Machinery, pp. 77–86.

[21] Schilit, B., Adams, N., and Want, R. Context-Aware Computing Applications. In *1994 First Workshop on Mobile Computing Systems and Applications* (Dec. 1994), pp. 85–90.

[22] Schmidt, A., Beigl, M., and Gellersen, H.-W. There is more to context than location. *Computers & Graphics 23*, 6 (Dec. 1999), 893–901.

[23] Schon, D. *Reflective Practitioner*. 1983.

[24] Wobbrock, J. O. Situationally aware mobile devices for overcoming situational impairments. In *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (Valencia Spain, June 2019), ACM, pp. 1–18.

[25] Yuksel, B. F., Oleson, K. B., Harrison, L., Peck, E. M., Afergan, D., Chang, R., and Jacob, R. J. Learn Piano with BACh: An Adaptive Learning Interface That Adjusts Task Difficulty Based on Brain State. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2016), CHI '16, ACM, pp. 5372–5384.

Jakob Karolus is a PhD student at the Human-Centered Ubiquitous Media lab at the Ludwig Maximilian University of Munich. Jakob received his master's degree in Visual Computing from TU Darmstadt in 2015. As part of his PhD project, Jakob focuses on establishing guidelines and techniques for proficiency-aware systems based on ubiquitous sensing technologies. His key interests lie in investigating opportunities and the design of engaging experiences for users to understand their own proficiency. He also conducts research in employing electromyography for sensory augmentation and novel interaction paradigms in human-computer interaction.

Address: LMU Munich, Computer Science Institute, D-80337 Munich, E-Mail: jakob.karolus@ifi.lmu.de

---

tekn. dr. Paweł W. Woźniak is an assistant professor for Human-Centred Computing at Utrecht University. Previously, he was postdoctoral fellow at the Chair for Human-Computer Interaction and Socio-Cognitive Systems, University of Stuttgart. Paweł received his PhD degree in Human-Computer Interaction from Chalmers University of Technology in 2016. Since then, he has been actively conducting research and supervising students. Paweł's key interests lie in the intersection of technologies, sport and wellbeing. His focus is on understanding the everyday experiences of physical activity to design better technologies that support wellbeing. He also conducts research in multi-surface interactions and augmenting sensory perception.

Address: Utrecht University, Computing and Information Sciences, 3584CC Utrecht, E-Mail: p.w.wozniak@uu.nl