Timing AI in HCI: Computational Approaches to Temporal Strategies for Mixed-Initiative Intelligent Systems

Daniel Buschek
Research Group HCI + AI,
Department of Computer
Science, University of Bayreuth
Bayreuth, Germany
daniel.buschek@uni-bayreuth.de

CHI'20,, April 25–30, 2020, Honolulu, HI, USA

Abstract
In this position paper, we revisit timing in mixed-initiative UIs through the lens of Computational Interaction, at the intersection of HCI and AI, to address the key question: *How can interactive AI realise good timing?* We observe that many current applications implement either fixed timing (e.g. always visible recommendations) or manual strategies (e.g. developer-specified user actions trigger AI actions). We propose instead to explore the computational view not just to inform system actions but also their timing – that is, to *learn* from interaction data how to realise task-specific temporal strategies for mixed-initative interfaces. We discuss examples of text generation and predictive text entry and suggestions for future research.

Author Keywords
Mixed-initiative; interactive AI; computational interaction

CCS Concepts
•**Human-centered computing** $\rightarrow$ **Human computer interaction (HCI);**

Introduction and Background
Mixed-initiative UIs sit between digital tools and agents by combining the former's direct manipulation concepts with automation capabilities provided by the latter, as described by Horvitz in 1999 [3]. In the same paper, he presents an

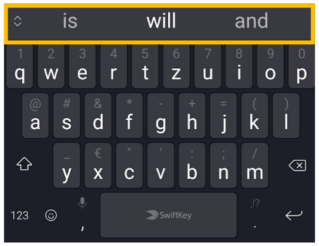

Figure 1: Current smartphone keyboards typically show word suggestions all the time (highlighted). This might not be ideal, considering the cognitive effort it takes typists to parse them [7]. As this position paper argues, *timing* such intelligent features well might be at least as important as improving their accuracy for current research at the intersection of AI and HCI.

example model for informing *when* an intelligent email assistant should suggest actions, e.g. adding a calendar entry. More generally, mixed-initiative of user and intelligent system implies that *timing* becomes a key consideration.

New Opportunities for Timing AI Engagements in a Task
Horvitz' timing model used email length, essentially for reading time estimates: Users spend more time with longer emails before they are ready to act, hence AI-suggested actions for each email could be delayed accordingly.

Twenty years later, we observe that this connects well to today's ideas of Computational Interaction [6], which also values such quantitative analyses of interaction data and models of user behaviour, including attention.

Revisiting timing through this lens is timely given manifold new opportunities in combining HCI and AI, including e.g. reinforcement learning, and wide-spread "intelligent" features, such as predictive text entry (Figure 1). Moreover, we now see more widely available potential information sources, such as eye tracking and physiological sensing, to gauge user attention and reactions to AI engagement.

In this position paper, we thus revisit timing in mixed-initiative UIs through the lens of Computational Interaction, at the intersection of HCI and AI.

Timing Mixed-Initiative in Creative Tasks
We highlight in particular a recent line of research which has explored mixed-initiative for applications in which the AI component contributes to *creative output*: For example, AI refines game levels sketched by human designers [9], while in the case of a computational art tool [1], human artists select design proposals generated by an AI. In general, supporting design tasks appears as a key use-case, including AI-supported creation of moodboards [5] and UI layouts [8].

Realising Temporal Strategies
Overall, timing in such examples is typically either realised

- by avoiding a temporal strategy altogether (e.g. always visible word suggestions on modern smartphone keyboards);
- by triggering AI engagement at fixed points in the interaction process (e.g. AI suggestion is updated when designer adds content [5]);
- via explicit user requests (e.g. "fix" button triggers local optimisation of the user's current UI sketch [8]);
- via turn-taking (e.g. each user action triggers new AI design evolutions [1, 9]).

We observe that such approaches require developers to manually select and implement strategies (e.g. which user actions exactly trigger AI). This likely limits timing to basic strategies. Moreover, it likely needs to be decided anew for each application, potentially explaining why many current AI deployments favour simple patterns, such as recommendation panels as fixed UI boundaries for intelligent features. This does not seem to realise the full potential of the vision of mixed-initiative UIs. Hence, here we explore how a computational, data-driven view might not just be applied to inform system actions but also their timing.

Computational Learning Paths
We next describe potential pathways for learning task-specific realisations of temporal principles of mixed-initiative UIs, using computational methods. We envision that these might allow a system to engage at the right moments, that is, when it is beneficial for the task, desired by the user, and without unwanted disruption. We illustrate these ideas with examples of AI-supported writing, which we seek to

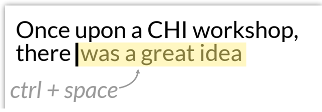

Figure 2: Example for timing AI engagement in a writing task without a learned strategy: The user **explicitly requests** text suggestions with a shortcut.

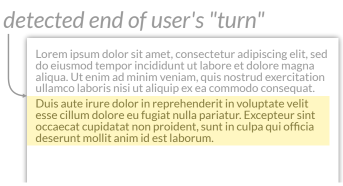

Figure 3: Example for timing AI engagement in a writing task by **learning to "take turns"**: The system has learned to recognise the end of the user's current writing process (e.g. idle time, semantic break, end of paragraph, gaze), and only then displays a possible continuation (yellow text).

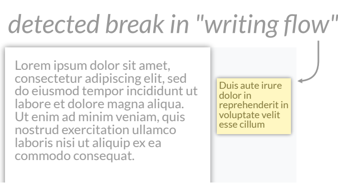

Figure 4: Example for timing AI engagement by **learning to minimise interruption**: The system has learned when to display suggestions in the margins (e.g. based on gaze data, attention on text vs margin, etc.). Suggestions could be added to the text e.g. via drag and drop.

address in our future work. This is a creative task in which the AI helps the user to compose a text, for example by suggesting sentence continuations (c.f. word suggestions, Figure 1; Google Smart Reply / Gmail [4]).

Zero Learning (e.g. User-Triggered, Always There)
The AI does not attempt to time its initiative right but rather waits for the user to explicitly request it. In our example, this could be a keyboard shortcut that triggers auto-complete suggestions (Figure 2). Related, a "fixed strategy" is also the current standard for word suggestions on smartphone keyboards – they live in a fixed bar, always visible (Figure 1). This can be seen as a baseline: In most applications any useful learned timing needs to be better in relevant (task) metrics than such explicit requests.

Learning to Take Turns
The system learns timing based on an assumption of turn-taking. In our example, it might learn to recognise that the user has finished writing and only then display suggested new text (Figure 3). This might be learned from semantic analysis of the text, its form (end of paragraph), behavioural measures like idle time, or even "activity" signals like EEG. It might be learned via implicit reinforcement (e.g. reward based on deletions of generated text) or explicit labels in an enrollment phase (e.g. "end turn" button).

Learning to Minimise Interruption
This approach builds on the prior assumption that interruptions cannot be avoided. In our example, such a system might assume that any text suggestion will interrupt the flow of the human writer. In terms of UI design it might thus add suggestions in the margins, not in the text body (Figure 4). To minimise disruption it might learn from behaviour and attention data (e.g. gaze location), using (implicit) feedback (e.g. negative reward: user stopped writing but ignored suggestion).

Learning to Maximise Utility
This approach uses the prior assumption that any timing can be acceptable as long as the AI's action has high utility for the user at that moment. Horvitz also described utility [3], yet with an emphasis on choosing appropriate actions – here, we envision it to also inform timing of said actions. In our example, such a system might suggest text when the user struggles with the wording, for example, learned from patterns of editing behaviour. It might also alert the user immediately if it detects inconsistencies (e.g. earlier statements, clashing calendar appointments).

Further Ideas
Further ideas include an emphasis on *user-specific* learned timing models, as well as the use of transfer learning, for instance, to learn timing based on data from other similar or related tasks, or other users in that task. In general, the presented pathways are in no way meant to be an exhaustive list and hopefully spark further ideas of how we might realise temporal principles in mixed-initiative UIs with a computational, data-driven perspective.

Discussion and Conclusion

Timing AI in HCI is important: Twenty years after Horvitz' mixed-initiative principles [3], recent guidelines for Human-AI interaction also state *"Time services based on context"* or *"Update and adapt cautiously"* [2]. While useful, such principles remain abstract until implemented.

To recap an example, current smartphone keyboards typically always show suggested words – which might not be ideal [7]. Even more accurate predictions might not help their usefulness. Instead, the key might be their timing.

To address such issues, we have argued for a closer look at how good task-specific timing of AI in HCI might be achieved. We have done so particularly from a viewpoint of Computational Interaction: This promises to base timing decisions on quantitative methods and models – much in the same vein as Horvitz' original email example.

To conclude, with today's computational, data-driven tools we should strive not only to inform which actions AI should take (or to improve their quality); we should rather also employ them to inform the *timing* of AI engagement in mixed-initiative tasks, for example, via models of user behaviour and attention, trained and working on interaction data.

REFERENCES

[1] 2015. Casual Creators. *Proceedings of the Sixth International Conference on Computational Creativity* June (2015), 8. http://axon.cs.byu.edu/ICCC2015proceedings/10.2Compton.pdf

[2] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, and et al. 2019. Guidelines for Human-AI Interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, Article Paper 3, 13 pages. DOI: http://dx.doi.org/10.1145/3290605.3300233

[3] Eric Horvitz. 1999. Principles of Mixed-Initiative User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99)*. Association for Computing Machinery, New York, NY, USA, 159–166. DOI: http://dx.doi.org/10.1145/302979.303030

[4] Anjuli Kannan, Karol Kurach, Sujith Ravi, Tobias Kaufman, Balint Miklos, Greg Corrado, Andrew Tomkins, Laszlo Lukacs, Marina Ganea, Peter Young, and Vivek Ramavajjala. 2016. Smart Reply: Automated Response Suggestion for Email. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD) (2016).* https://arxiv.org/pdf/1606.04870v1.pdf

[5] Janin Koch, Andrés Lucero, Lena Hegemann, and Antti Oulasvirta. 2019. May AI? Design Ideation with Cooperative Contextual Bandits. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, Article Paper 633, 12 pages. DOI: http://dx.doi.org/10.1145/3290605.3300863

[6] Antti Oulasvirta, Per Ola Kristensson, Xiaojun Bi, and Andrew Howes (Eds.). 2018. *Computational Interaction*. Oxford University Press.

[7] Philip Quinn and Shumin Zhai. 2016. A Cost-Benefit Study of Text Entry Suggestion Interaction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. Association for Computing Machinery, New York, NY, USA, 83–88. DOI:http://dx.doi.org/10.1145/2858036.2858305

[8] Kashyap Todi, Daryl Weir, and Antti Oulasvirta. 2016. Sketchplore: Sketch and Explore with a Layout Optimiser. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems (DIS '16)*. Association for Computing Machinery, New York, NY, USA, 543–555. DOI: http://dx.doi.org/10.1145/2901790.2901817

[9] Georgios N. Yannakakis, Antonios Liapis, and Constantine Alexopoulos. 2014. Mixed-Initiative Co-Creativity. In *Proceedings of the 9th Conference on the Foundations of Digital Games*.