Generalizing User Models through Hybrid Hierarchical Control

Thomas Langerak
thomas.langerak@inf.ethz.ch
ETH Zürich
Switzerland

Sammy Christen
sammy.christen@inf.ethz.ch
ETH Zürich
Switzerland

Anna Maria Feit
feit@cs.uni-saarland.de
Saarland University, Saarland Informatics Campus
Germany

Otmar Hilliges
otmar.hilliges@inf.ethz.ch
ETH Zürich
Switzerland

ABSTRACT

Reinforcement Learning has two main challenges in the field of Human-Computer Interaction. The first challenge is generalization across tasks and environments. The second challenge is to achieve human-likeness. We propose a Hybrid Hierarchical Control framework for pointing tasks to address both challenges simultaneously. In our framework, we separate high-level decision-making from low-level motor and gaze control. This hierarchical structure promotes generalizability. By constraining the low-level control to human-like capabilities we aim to achieve human-like results. Finally, we present some applications that our framework could be used for.

KEYWORDS

User Interface Optimization, User Modeling, Reinforcement Learning, Hierarchical Control

ACM Reference Format:
Thomas Langerak, Sammy Christen, Anna Maria Feit, and Otmar Hilliges. 2021. Generalizing User Models through Hybrid Hierarchical Control. In *Proceedings of CHI 2021 Workshop on Reinforcement Learning for Human-Computer Interaction (RL4HCI@CHI '21).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

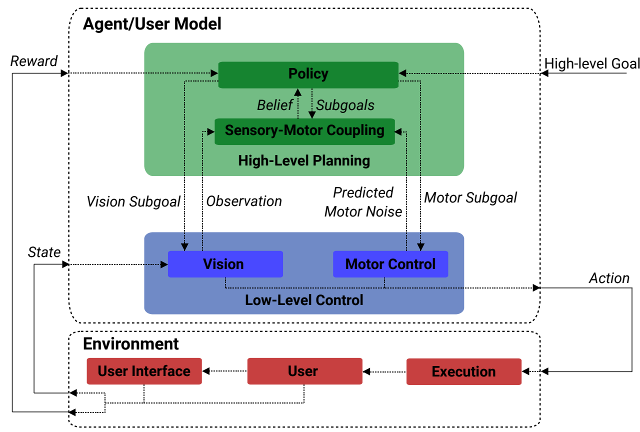

Figure 1: Overview of our proposed Hybrid Hierarchical Control framework. The agent (a user model) interacts with the environment (user interface and user) through actions. These are decided on by partial observations of the environment state. We decompose the agent into hierarchical layers separating high-level planning from low-level control.

1 INTRODUCTION

Recent advances in Reinforcement Learning (RL) have paved the way for its use in Human-Computer Interaction (HCI), offering researchers unique ways to model sequential decision-making in humans and machines [2, 6]. While RL can been used to optimally adapt user interfaces [4, 14] it shows particular promise for modeling the behavior of users when interacting with (intelligent) interfaces. In particular, RL has been successfully used to explain task-interleaving behavior [5], to predict movements for mid-air interaction [1], and to model motor and gaze behavior during soft keyboard typing [8].

RL4HCI@CHI '21, May 08, 2021,

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Despite these successes, we see two important issues that make RL challenging to apply to HCI problems: (1) the ability to generalize to unseen tasks or environments and (2) the ability to exhibit human-likeness. A problem of RL models is that they easily overfit to the training task. This leads to poor performance when the environment or reward function change, a common case when interacting with computers. For example, as humans we can easily switch from entering a phone number on a touchscreen to pressing the physical keys on a landline phone. In such a case, our input strategy stays the same and we only adapt our motor behavior. This is not straightforward for an RL model trained on the task of operating a touchscreen. Generally, as HCI researchers, we are interested in evaluating different user interfaces (UIs) and explore variations in users' behavior. In these scenarios, retraining a policy for each task is tedious and computationally expensive. Therefore, a policy that generalizes across tasks is required.

The second issue arises when RL optimizes a hand-crafted reward function, which may not lead to a *human-like* policy. To overcome this challenge, Inverse Reinforcement Learning (IRL) [10] or Imitation Learning [12] can be used, which do not rely on a hand-crafted reward function, but learn from demonstration data. However, this is not a satisfying solution, as the demonstration data is costly and time-consuming to collect and, again, is difficult to generalize to different users or environments. Altogether, for Reinforcement Learning to become successful in HCI, the learned model's *generalization* capabilities and its similarity to humans need to be increased.

We propose a *Hybrid Hierarchical Control* framework as a solution to both problems. Our method is inspired by Hierarchical Reinforcement Learning, which splits a decision making problem into several subgoals at different levels of abstraction [13]. Following prior work in robotic control and navigation [3], our framework goes even further and introduces an explicit hierarchical *separation of states and actions* into low-level control and global planning.

The low-level control uses existing approaches for modeling aspects related to the human body that govern our ability to interact with a UI, such as motor-control or perception. These are mostly well-studied and do not necessarily need to be learning-based. The hig-level planner then learns a policy that distributes subgoals to these components in order to solve a specific task. Such strategic planning is typically less understood and varies by task.

This separation has two advantages. (1) Modularity and generalization: it allows to combine different planning and control modules. The lower level can generalize across tasks. For example, the motor control for pressing a button on a touch-screen stays roughly the same independent of the specific UI and task. On the other hand, the same high-level planner could be combined with different control models capturing variations in motor and vision capabilities (e.g. colorblindness or tremor) or different input modalities. (2) Promoting human-like policies: Since the different layers are functionally decoupled, we can benefit from known models of human motor control and perception that do not need to be learned. This constrains the model to human-like policies without the need for demonstration data. It allows us to explain interactive behavior as the result of optimal decision making bounded by the constraints of the human body and task requirements [2, 6].

In the following, we demonstrate how we realize this separation of concerns to model interactive behavior that involves one-finger pointing. Pointing is an ubiquitous part of many computer interaction tasks and includes decision-making, vision, and motor control. The pointing problem is heavily resource-constrained (e.g., one cannot look everywhere at the same time, decisions are not reached in an instant and limbs do not have infinite velocity). At every point in time, the user has to decide how to manage these resources optimally, allowing us to make use of the rationality assumption [9]. This makes it an interesting and highly relevant case.

2 FRAMEWORK

The goal of our work is to develop a user model that generalizes across tasks while maintaining human-likeness. We focus on interactive tasks that involve one-finger pointing. This encompasses a large number of use-cases including operating many button-based interfaces e.g. on mobile touch screen devices, public displays, or for mid-air interaction.

We propose a Hybrid Hierarchical Control framework that separates task-dependent planning from internal control, as shown in Figure 1. This captures an aspect inherent to many interaction scenarios: the user has to make decisions about how to operate an interface without having full knowledge about its state. To gather information about it, the user has to optimally manage the limited vision and motor resources which are also used to manipulate the interface.

The high-level planning models the task-oriented cognition (Sec 2.1) which provides subgoals (the policy) for the lower-level components in the hierarchy. It has no direct access to the environment, which includes the state of the user (e.g. their finger and gaze position) and the user interface the agent is interacting with. Instead, it integrates the sensory feedback (e.g., haptic and visual cues) provided by the low-level control to update its belief about the environment state (sensory-motor coupling).

The lower level comprises the motor (Sec. 2.2.1) and vision (Sec. 2.2.2) control, which are guided by the subgoals provided by the higher level. The lower level interacts directly with the environment via control actions, changing both the user and, thereby, the user interface state. The proprioceptive state of the motor or visual system is only available to the respective low-level controller

2.1 High-Level Planner

Following prior work [2, 6], we model the high-level planner as a Partially Observable Markov Decision Process (POMDP). As a POMDP, the decision making module does not have direct access to the environment state. Instead, it receives observations from the lower layers, such as visual feedback. These are combined in the sensory-motor coupling submodule to infer a belief state of the environment. The belief state encompasses both the believed state of the user (e.g., the end-effector position) as well as the state of the user interface (e.g., position of the next pointing target), while omitting information about the internal state of the user (e.g. specific muscle activations). We aim to learn a policy that takes optimal actions given the belief state and the high-level goal, in order to maximize an underlying environment reward. Examples for high-level goals could be to hit a moving target in a game or to type a certain word on a keyboard. We can find this policy by solving the POMDP problem through Deep Reinforcement Learning.

2.2 Low-level Control

The low-level control comprises the motor control and vision. They execute actions in the environment and provide observations to the high-level planner.

2.2.1 Motor Control. The motor module controls the finger location of the agent. It is used to manipulate the interface and provides information about the motor noise to the high-level planner. In line with previous work, we model the motor control as a Model Predictive Control optimization problem [7]. Given a pointing target (the subgoal provided by the high-level planner), it computes a series of optimal force activations of the user's limb by minimizing the control actions $\mathbf{u}$ (force) over a time horizon N, such that the target is reached at the final step. We approximate the limb as a mass-point model. The control actions are optimized based on the belief of the high-level planner about the current position of the finger. This might deviate from the actual state of the environment where the planned movement is then executed. For execution, we add Gaussian noise to the inputs. The variance of the Gaussian noise scales linearly with the inputs applied (i.e., movements that require more force are noisier). The average force is influenced by both movement time and distance, as distance traveled is a function of time and force. Our distance is known ahead of time, since we have access to the target and know it should be reached at the final step of the horizon. Hence, the variance is implicitly influenced by movement time (i.e. horizon length). Due to this we are able to model the speed-accuracy trade-off as a weighted optimization problem to inform the choice of the time horizon N. This trade-off can be influenced by the high-level planner in terms of weights.

2.2.2 Vision. The vision module controls the eye gaze of the agent. It is used to obtain visual information about the environment (the state of the user interface and that of the motor system) and thus reduce the uncertainty in the high-level decision making. To simulate the user's eye gaze, we rely on prior work by Salvucci [11], who introduced the EMMA (eye movements and movement of attention) model. Given a UI element, the model predicts the temporal and spatial characteristics of the eye movements needed to attend to that object and visually encode it. In doing so, it separates the covert attention of the user from their eye movements, dividing the process into two components that run in parallel: attending and visually encoding an object, and moving the eyes as a result of the attention shift. With that, it takes into account that information might be perceived in the peripheral view and encoded without moving the eyes. See the original work for more details [11].

Practically speaking, the EMMA model takes as input a target location (the subgoal provided by the high-level planner) and outputs information about the environment at the given location to the high-level planner. As part of this it computes a series of (noisy) gaze positions, if any, and the corresponding movement times. This includes the time it takes to visually encode the attended element, even if no eye movement was performed The simulated eye movements are executed in the environment to change the gaze position of the user.

3 DISCUSSION

We present a Hybrid Hierarchical Control framework to model user behavior for pointing interactions. Our hypothesis is that the hierarchical separation into local control and global planning allows for generalization across different interfaces, input modalities, and user characteristics. In order to validate these hypotheses, we will, implement the proposed framework and validate its generalizability and human-likeness on different scenarios where either the task stays the same and the input modality varies, or the other way around. A key open question here is how general the reward for the high-level planning can be defined.

3.1 Envisioned Applications

Current UI optimization and design approaches require models or heuristics of user behavior that are tailored to the characteristics of the specific task and environment. Our proposed framework allows to create general user models that can be used to evaluate or optimize a variety of pointing-based user interfaces. For example, the same model could be used to optimize a webpage on a 2D screen or optimally arrange interactive elements in an augmented reality application (with respect to the user's pointing behavior). In both cases, the high-level behavior of a user emerges as a result of the low-level control and the task-dependent reward. It does not need to be specified or empirically measured for the specific task.

Similarly, an RL-based model can capture changes in the user's behavior as a response to variations in the design. Therefore, we can optimize for aspects that were not possible with static models. For example, when designing a word prediction algorithm, the number of words shown to the user might change their optimal behavior (e.g. whether they attend the list or not). With an RL-based model as proposed here, we are able to capture this variation and take it into account during optimization.

Another application case is user state prediction over a horizon. Interactions can be greatly improved if we know user trajectories. However, to achieve this, we need to be able to accurately predict user behavior given the current state. Our model is capable of this by initializing it with the user's and UI's current state and unrolling future states.

Finally, it is straightforward to approximate different users. For instance, by increasing noise levels on the motor controller, we may be able to approximate Parkinson's disease. One would expect that this results in a different optimal high-level policy. An interesting avenue for future work is to investigate the interplay between the high-level planner and varying low-level controllers.

3.2 Challenges and opportunities

Our proposal is not without challenges. When evaluating a model, it is important to find a metric that correctly assesses the desired properties. This is even more of a concern when developing models of human behavior. For this, we need metrics that capture 'human-likeness'. This could range from simple distribution metrics to neural networks that try to differ between real and synthetic data (similar to a discriminator in General-Adversarial Networks).

Another challenge is that the high-level planning needs a reward function, which can take many forms. For example, the task completion time, error rate, or a weighted sum of both could be used. How a suitable reward function can be formulated that includes also more subjective criteria needs further investigation.

In our proposal, we specifically focus on motor control and vision as limited resources that constrain interactions. However, due to the modular nature of the framework, this can easily be extended to include additional components such as auditory cues or haptic perception. These extra cues might provide additional means to update the belief state and thus result in more human-like policies. It would allow us to investigate the effect of multimodal interfaces.

3.3 Conclusion

To succeed in human-computer interaction, specifically in user modeling, RL needs to overcome the challenges of generalizability and human-likeness. We propose to address both via a Hybrid Hierarchical Control framework. The hybrid nature, mixing reinforcement learning with classical strategies, could promote human-likeness, since we can leverage existing models with human-like constraints to the system. Furthermore, the functional decomposition, achieved via explicit separation of the hierarchical levels, increases generalizability.

REFERENCES

[1] Noshaba Cheema, Laura A Frey-Law, Kourosh Naderi, Jaakko Lehtinen, Philipp Slusallek, and Perttu Hämäläinen. 2020. Predicting mid-air interaction movements and fatigue using deep reinforcement learning. In *CHI*. 1–13.

[2] Xiuli Chen, Sandra Dorothee Starke, Chris Baber, and Andrew Howes. 2017. *A Cognitive Model of How People Make Decisions Through Interaction with Visual Displays*. ACM, New York, NY, USA, 1205–1216.

[3] S. Christen, L. Jendele, E. Aksan, and O. Hilliges. 2021. Learning Functionally Decomposed Hierarchies for Continuous Control Tasks with Path Planning. *IEEE Robotics and Automation Letters* (2021).

[4] Christoph Gebhardt, Brian Hecox, Bas van Opheusden, Daniel Wigdor, James Hillis, Otmar Hilliges, and Hrvoje Benko. 2019. Learning Cooperative Personalized Policies from Gaze Data. In *UIST* (New Orleans, LA, USA). ACM, New York, NY, USA, 197–208.

[5] Christoph Gebhardt, Antti Oulasvirta, and Otmar Hilliges. 2020. Hierarchical Reinforcement Learning Explains Task Interleaving Behavior. *Computational Brain & Behavior* (nov 2020), 1–21.

[6] Andrew Howes, Xiuli Chen, Aditya Acharya, and Richard L. Lewis. 2018. Interaction as an emergent property of a partially observable Markov decision process. In *Computational Interaction*. Oxford University Press, 287–310.

[7] Yifeng Jiang, Tom Van Wouwe, Friedl De Groote, and C Karen Liu. 2019. Synthesis of biologically realistic human motion using joint torque actuation. *ACM Transactions On Graphics (TOG)* 38, 4 (2019), 1–12.

[8] Jussi P P Jokinen, Aditya Acharya, Mohammad Uzair, Xinhui Jiang, and Antti Oulasvirta. 2021. Touchscreen Typing as Optimal Supervisory Control. In *CHI*. ACM.

[9] James G March. 1978. Bounded rationality, ambiguity, and the engineering of choice. *The bell journal of economics* (1978), 587–608.

[10] Andrew Y. Ng and Stuart Russell. 2000. Algorithms for Inverse Reinforcement Learning. In *ICML*. Morgan Kaufmann, 663–670.

[11] Dario D Salvucci. 2001. An integrated model of eye movements and visual encoding. *Journal of Cognitive Systems Research* 1 (2001), 201–220.

[12] Stefan Schaal. 1999. Is imitation learning a route to humanoid robots? *Trends in cognitive sciences* 3 (07 1999), 233–242.

[13] Richard S. Sutton, Doina Precup, and Satinder Singh. 1999. Between MDPs and semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning. *Artif. Intell.* 112, 1-2 (Aug. 1999), 181–211.

[14] Kashyap Todi, Luis A Leiva, Gilles Bailly, and Antti Oulasvirta. 2021. Adapting User Interfaces with Model-based Reinforcement Learning. (2021).