Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 10, No. 2, 242-259
2026
Publisher: Learning Gate
DOI: 10.55214/2576-8484.v10i2.12015

A human-centered deep reinforcement learning framework for user experience–driven personalized user interface generation

Ahmed Alshehri[1*]
[1]Department of Information Technology, Faculty of Computing and Information, Al-Baha University, Al-Baha, Saudi Arabia; a.alyehyawi@bu.edu.sa (A.A.).

Abstract: Digital systems are becoming increasingly feature-rich and interaction-intensive, while users vary widely in their goals and expertise, cognitive styles, accessibility requirements, and device contexts. This diversity makes one-size-fits-all interfaces inefficient and, at times, frustrating, often increasing error rates, cognitive load, and user dissatisfaction. Existing personalization approaches, such as themes, fixed preferences, and rule-based customizations, offer limited flexibility and fail to adapt to evolving user behavior and contextual changes. Although AI-driven adaptive interfaces have shown improvements, most approaches remain system-centric and insufficiently address human-centered considerations. This often results in disruptive interface changes, a perceived loss of control, and diminished user trust. This paper proposes a Human-Centered Deep Reinforcement Learning (HC-DRL) framework for generating personalized user interfaces, in which UI adaptation is modeled as a constrained sequential decision-making process. The framework combines continuous user modeling with a structured representation of the user interface based on a design system. A DRL agent predicts viable adaptation policies using a UX-sensitive reward function that explicitly maximizes task success and efficiency while accounting for user satisfaction, cognitive load, trust, perceived control, and disruption penalties. Safety guardrails are incorporated to enforce accessibility and usability constraints and to enable rollback to stable interface states when risks or performance degradation are detected. An end-to-end implementation and evaluation pipeline, including comparisons with static and heuristic baselines, ablation studies to quantify component contributions, and user studies, was employed to validate the proposed approach. The results demonstrate that HC-DRL provides a practical and robust foundation for adaptive interfaces that enhance functionality without compromising stability, accessibility, or user confidence.

Keywords: *Adaptive interfaces, Deep reinforcement learning, Human-centered reinforcement learning, Personalized user interfaces, User experience (UX), UX-aware interface adaptation.*

1. Introduction

Digital systems today are becoming more and more marked by an overgrowth of features and interactivity modalities, and thus more complex in how they are structured, navigated, and the decision-making processes. At the same time, the groups of users who interact with such systems are highly diverse and vary in their objectives, knowledge, thinking habits, accessibility needs, device conditions, and contextual limitations [1]. The resulting mismatch between universal, so-called one-size-fits-all interfaces and the differentiated user requirements often results in inefficient task performance, increased rates of error, cognitive overload, and lower levels of satisfaction [2]. Although traditional methods of personalization, such as fixed themes, manually defined preferences, or rule-based customizations, can provide only limited customization because they are not typically capable of dynamically adjusting to changing user behavior and contexts [3]. As a result, the need to devise smart

History: Received: 24 November 2025; Revised: 14 January 2026; Accepted: 19 January 2026; Published: 4 February 2026
* Correspondence: a.alyehyawi@bu.edu.sa

UI systems that can constantly learn based on interaction cues and produce customized UI settings that optimize performance as well as user experience is irresistible.

Although recent advances have made AI-based personalization, an apparent gap still exists [4-6]. Many adaptive UI methodologies focus on system-centric performance (i.e., clicks, completion rates, or time-on-task) without considering user experience as a secondary aspect [7]. This tendency of excessive adaptation may breed unexpected changes in the interface, perceived loss of control, and loss of trust, especially in real-world situations where comfort, clarity, and usability are equally important [8]. Deep Reinforcement Learning (DRL) is a conceptual model for modeling the problem of UI personalization as a sequence of steps, in which the agent can learn optimal adaptation policies through interactive and feedback mechanisms [9]. However, the formulations of standard Deep Reinforcement Learning for Adaptation (DRA) are not necessarily human-centered unless user experience (UI) aspects are explicitly incorporated into the learning task and constraint mechanisms; otherwise, the emerging UI behaviors are not necessarily human-desirable [10]. In this regard, the paper presents a human-centered DRL framework to personalize the creation of UI and explicitly relate the learning process to usability, cognitive comfort, satisfaction, trust, and perceived control.

This work aims to design, deploy, and test a Human-Centred Deep Reinforcement Learning (HC-DRL) system to generate personalized UI that (i) would learn the adaptive policies of the UI through user interaction data and environmental features, (ii) would directly incorporate human-centered metrics into the reward design and guardrails, and (iii) would provide explainable, stable, and safe implementations of the UI that would not compromise the usability and autonomy of the user. The framework suggested puts together user modeling (capturing preferences and behavior), structured UI representation (capturing layout and component constraints), a DRL agent to make sequential personalization choices, and a UX-sensitive reward signal, a combination of task success and efficiency with subjective and cognitive experience signals.

This study is guided by four research questions. RQ1 questions the hypothesis that a DRL-based personalization agent can produce better UI configurations in terms of task performance compared to the baselines (static and rule-based). RQ2 questions whether the inclusion of human-centered UX indicators in the reward mechanism can produce interfaces that are perceived as more usable, more trustworthy, and controllable compared to those optimized for task performance. RQ3 explores how far the proposed framework can be generalized among different users, tasks, and contexts of interaction without provoking disruptive and inconsistent changes in the UI. RQ4 assesses how the main components, i.e., user modeling, UX-integrated reward shaping, and constraint-based guardrails, contribute to overall performance and UX results. Thus, it hypothesizes that UX-congruent DRL will increase task success and efficiency and, at the same time, increase satisfaction and perceived control, decrease cognitive load, and increase trust. Usability guardrails will additionally alleviate stability and decrease negative adaptation events.

The key contributions outlined in this paper consist of four distinct dimensions: (1) a human-centered definition of deep reinforcement learning to generate personalized user interfaces, formalized as a stepwise decision-making process integrating explicit compliance with human experience, (2) a user-experience-based reward strategy that balances both task success and operational efficiency with satisfaction, cognitive load, trust, and perceived control, (3) a constraint-sensitive interface adaptation mechanism, which imposes usability and accessibility guardrails that ensure unsafe or disruptive interface changes are prevented, and (4) a complete end-to-end framework and evaluation pipeline with baselines and ablations that jointly measure interaction performance and human-centered UX outcomes.

The rest of this paper is organized as follows: Section 2 reviews related work. Section 3 formulates the problem and assumptions. Section 4 presents the proposed HC-DRL framework. Section 5 describes the implementation and evaluation methodology. Section 6 reports results. Section 7 discusses implications and limitations, and Section 8 concludes the paper with future research directions.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 10, No. 2: 242-259, 2026
DOI: 10.55214/2576-8484.v10i2.12015

2. Related Work

Using DRL to create personalized user interfaces (UIs) that can learn from each user's unique preferences is a current topic in the human-centered design field today. The DRL system, as defined by Lv et al. [11], uses user feedback about an image to guide the program when improving and classifying that same image. The result is a personalized distribution of images that fit with what a user prefers aesthetically. As the user provides direct feedback to the system during this process, the framework represents the viability of using a user-guided approach to help improve personalization in the process of creating UIs. In addition to exploring how to personalize interaction with the User Interface, researchers have begun to investigate how to apply DRL for dynamic decision-making in complex environments. Authors Du et al. [12] proposed a diffusion-based Reinforcement Learning framework to identify which AI-Generated Content Service Provider (AI-GCSP) is optimal given the variability and uncertainty of the environment in which they operate. While the work focuses on identifying the best AI-GCSP, it reflects the general capability of DRL systems to provide personalized decision support under uncertainty and is relevant to Adaptive UI creation. Authors Xu et al. [13] and Xu et al. [14] advocate for the use of federated deep reinforcement learning schemes to improve the deployment of Unmanned Aerial Vehicles (UAVs) within 6G networks, while maximizing throughput and privacy. Due to their unique capacity to continuously adapt to the state of the network and provide personalized experiences for individual users, these DRL schemes illustrate how the adaptive capabilities of Federated DRL can help create more personalized UIs. In the area of Federated Learning [15], a new elastic federated learning algorithm leverages adversarial autoencoders to create better models based on data differences between users, thus improving the accuracy of models created using federated learning. This method enables a personalized and privacy-preserved experience for users, which is critical for developing user-centric UI systems that adapt while maintaining user privacy. A study by Li et al. [16] provides additional research on individual differences in user preferences through language modeling and describes methods for creating personalized language models based on human feedback, emphasizing the need to accommodate user differences within a personalized UI system. Recently, researchers have explored how AI and humans work together as partners to increase the intelligence capabilities of both systems through technological advances. SymbioticRAG [17] uses logs of user interactions to understand user intentions. By using a two-way learning method, this system also shows how feedback from users can and should be used to improve the output of AI systems. This feedback can help AI systems adapt their user interfaces based on ongoing interactions with users. Another example of how user collaboration creates more accurate recommendations is the hybrid recommendation framework developed by Diao et al. [18], which includes transformer-based models, reinforcement learning, and contrastive self-supervised learning (all enabling the prediction of user behavior) into a hybrid recommendation framework. This approach uses multiple modes of learning and adapting to utilize diverse data sources, demonstrating that combining various methods for providing users with personalized recommendations and experiences creates a higher level of personalization, which can inform UIs. In recent research, Yang and Wang [19] illustrate how providing immediate user feedback about their actions can lead to a better user experience, as shown by the high satisfaction levels of users of their AR-based coaching program. A study by Feng and Jiang [20] describes how providing summarized reviews that mirror the user's persona creates a customized recommendation, thereby increasing user engagement in online shopping.

The literature review clearly demonstrates that while DRL, federated learning, and human-in-the-loop techniques support personalization and adaptive decision-making, the current body of literature on these methods tends to be primarily system-centric and domain-dependent. To date, there has been little research on how the explicit inclusion of key human-centered UI principles, usability, cognitive load, trust, accessibility, and perceived control, can be incorporated into the design of the states, rewards, and constraints used in DRL-based UI adaptation frameworks.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 10, No. 2: 242-259, 2026
DOI: 10.55214/2576-8484.v10i2.12015

3. Problem Formulation and Scope

This study outlines the extent of customized UI generation and defines the learning problem as a serial decision-making process, making it highly suitable for reinforcement learning methods. The personalized UI generation involves the automatic creation of a UI instance and its adaptation to a specific user within a given design space. The goal is to improve task execution and provide a human-centric user experience over time.

UI generation is construed pragmatically as a synthesis of a UI structure by the selection, positioning, and parameterization of elements in an existing design system, in place of the synthesis of a completely new visual language or application functionality. Changes in the layout structure and information hierarchy, rearrangement of the UI component, are a choice between functionally equivalent controls and the display parameters setting. Fine-tuning of interaction mechanisms, such as guidance levels, confirmation prompts, and progressive disclosure, is included as a customization aspect within the scope of this framework. This proposed framework provides a means to create personalized UI design by utilizing constrained optimization, wherein user and contextual variables are used to define the user experience of the application, while ensuring that the same functional, legal, and safety constraints applied within the framework are also enforced to create a user-specific UI design.

3.1. Personalized UI Generation definition

Individual UI generation can be termed as the generation of a user-specific interface configuration that will constantly be modified through time as the system monitors the interaction behavior and contextual context. The customization is carefully limited to presentation-level, structure-level, and interaction-level decisions that do not change the underlying semantics and functionality of the application. In particular, the structure promotes changes in the structure of layout, prioritization of information, the process of choosing functional equivalents in the widgets, the visual density and readability values, and the amount of guidance or friction involved in interaction processes. Concurrently, core system properties are fixed. These are important application features, task definitions, and correctness criteria, compliance-related content, and security or safety-critical functions, including authentication and irrevocable operations. The given definition is aimed at a careful balance between flexibility and stability with the aim of making sure that it will not negatively impact the usability and performance of the system, as well as user trust, predictability, and regulatory compliance.

3.2. System Assumptions

To design a UI that works on all device types (desktop, tablet, and mobile) and allows the system to store the necessary context about the user and the methods they use to interact with the interface. To implement this idea, we need to create a design system that consists of standard templates and elements so that the personalization engine can produce a user-specific version of the interface that is both functionally correct and visually similar. The system logs the interactions of the user with the interface through rich interaction traces (i.e., click-throughs, scroll depth, dwell time, failed attempts, backtracking, task completion), which provide feedback for the personalization engine's learning. This allows the personalization agent to improve its learning by updating its policy through both real-time and batch retraining options. Additionally, while explicit user feedback in the form of ratings or surveys may also be collected, a basic interface is included as a safety net for the user when either violations of constraints are detected or the personalization engine does not have high confidence in the adaptation created.

3.3. Constraints and Guardrails

To meet the needs of users while maintaining a high degree of end-user productivity, developers must design interfaces that provide human-centered personalization and ensure not to undermine end-user accessibility, responsiveness, usability, or safety. Accessibility constraints help promote inclusivity by requiring that typography is readable, contrast levels are adequate, and target sizes are appropriate

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 10, No. 2: 242-259, 2026
DOI: 10.55214/2576-8484.v10i2.12015

for fingers. The interface can be navigated with a keyboard and assistive technology. Responsiveness constraints support the legibility and functionality of generated pages across devices of different screen sizes and minimize the additional latency introduced as a result of adapting the interface. Usability limitations are determined by generic usability heuristics and include consistency, error prevention, and providing clear feedback to the user. When such actions are taken, and minimal surprises are performed when interacting with the interface, they consequently limit both the magnitude and frequency of changes to the interface to minimize confusion and loss of control by users. Safety constraints restrict the maximum amount of change and frequency of change to security-sensitive or irreversible user workflows and provide rollback capabilities. Thus, they limit the rate at which the interface can be changed and maintain conservative behavior in the presence of uncertainty. Combining these requirements creates an environment in which forbidden configurations (i.e., hard feasibility constraints), as well as tools to create overly disrupted (i.e., soft penalty mechanisms) but still technically feasible configurations, exist.

3.4. MDP/POMDP Formulation of Personalized UI Generation

Personalized UI generation is modelled as a Markov Decision Process [21] (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where the environment is the interaction loop between a user and the interface over a task or session. At each step t, the agent observes a state $s_t \in \mathcal{S}$ capturing user, context, UI, and recent behavior, commonly written as:

$$s_t = [\mathcal{U}_t, c_t, x_t, h_t] \quad (1)$$

where $\mathcal{U}_t$ is the user model, c_t, is context (device/task), x_t is the current UI configuration, and h_t summarizes recent interaction signals. The agent selects an adaptation action $\alpha_t \in A(st)$ from a state-dependent feasible action set constrained by design validity and guardrails, using a policy

$$\pi_\theta(a_t \mid s_t) = Pr(a_t \mid s_t; \theta) \quad (2)$$

The environment transitions according to

$$s_{t+1} \sim \mathcal{P}(.\ s_t, a_t), \quad (3)$$

where UI updates are often deterministic $x_{t+1} = \mathfrak{g}(x_t, a_t)$ while user responses and user-model updates remain stochastic.

The reward integrates task and human-centred outcomes. A compact formulation is

$$\mathfrak{r}_t = a\mathfrak{r}_t^{task} + \beta\mathfrak{r}_t^{eff} + \gamma\mathfrak{r}_t^{ux} + \lambda\mathfrak{r}_t^{disrupt} \quad (4)$$

with a typical instantiation

$$\mathfrak{r}_t = \alpha Succ_t - \beta Time_t - \eta Err_t + \gamma Sat_t - \delta Load_t + \mathcal{K}trust_t - \lambda d(x_t, (x_{t+1}), \quad (5)$$

Where $d(x_t, (x_{t+1})$ measures UI- change disruption. The learning objective is to maximize expected discounted return,

$$J(\theta) = \mathbb{E}_{\pi\theta}[\sum_{t=0}^{t} \gamma^t r_t] \text{ with value functions} \quad (6)$$

When strict usability/accessibility limits are required, constraints can be expressed as

$$max_\pi\ E[\sum_{t=0}^{T} \gamma^t r_t]\ s.t.\ E\left[\sum_{t=0}^{T} \gamma^t\ {c_t}^{(k)}\right] \le \xi_k \quad (7)$$

or handled via a Lagrangian reward

$$\tilde{r}_t = r_t - \sum_k^n \lambda_k\ {c_t}^{(k)} \quad (8)$$

Because latent variables such as intent, frustration, and workload are not fully observable, the problem can be modeled as a Partially Observable Markov Decision Process (POMD)P $\langle S, A, P, R, O, Z, \gamma \rangle$ where observations satisfy

$$o_t \sim Z(\cdot \mid s_t) \quad (9)$$

The agent may maintain a belief state

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 10, No. 2: 242-259, 2026
DOI: 10.55214/2576-8484.v10i2.12015

$$b_t(s) = Pr(s_t = s \mid o_{0:t}, a_{0:t-1}). \quad (10)$$

or approximate hidden-state inference using memory-based policies,

$$m_t = f(m_{t-1}, o_t), \; a_t \sim \pi(\cdot \mid m_t). \quad (11)$$

Equations (1-3) model personalized UI adaptation as a sequential decision process by defining the interaction state, policy, and state transitions. Equations (4-6) specify a human-centred objective that balances task performance, efficiency, user experience, trust, and disruption over time. Equations (7-8) incorporate usability and accessibility constraints, while Equations (9-11) address partial observability of user states through belief or memory-based policies for robust adaptation.

4. Proposed Human-Centred DRL Framework

4.1. Framework Overview and Architecture

The human-centered deep reinforcement learning (HC-DRL) paradigm aims to create and optimize user interfaces through a design architecture that treats UI adaptation as a sequential decision-making problem, with task performance and user experience being co-optimized. The architecture functions as a closed feedback system, as shown in Fig 1. It begins with rendering a UI configuration, after which the user interacts with the interface. Interaction signals are captured, enabling the learning agent to adjust its personalization policy to better meet user needs while balancing usability and safety principles. The system comprises four closely linked layers. The sensing layer records interaction events, such as clicks, scrolling behavior, dwell time, errors, backtracking, and task completion signals, along with contextual variables like device type, viewport characteristics, input modality, time, and task category. The user understanding layer then converts these raw signals into a structured user model, capturing both stable properties (e.g., accessibility preferences, expertise development) and dynamic conditions (e.g., inferred hesitation or confusion).

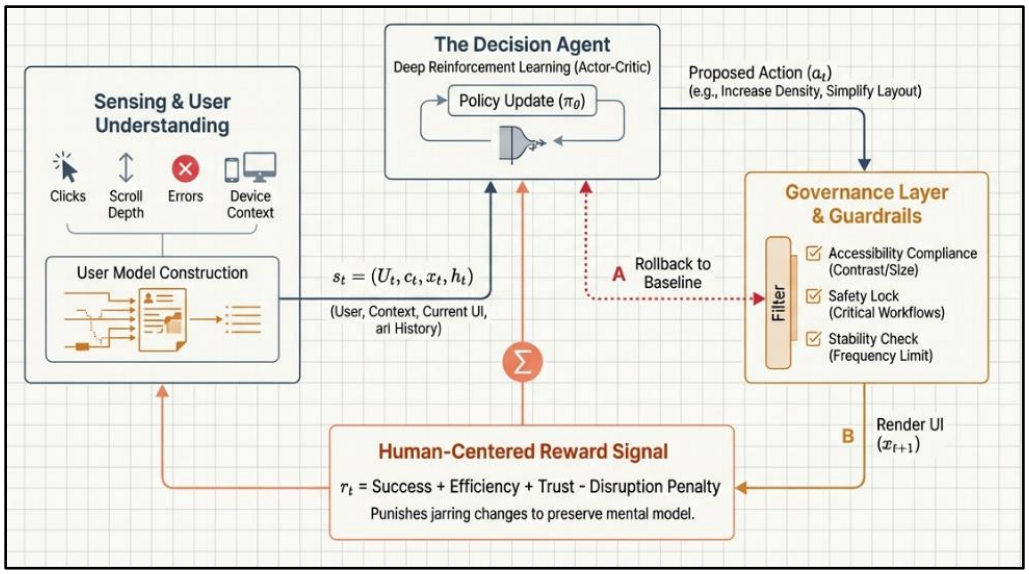

Figure 1.
Overview of the Proposed Architecture.

A deep reinforcement learning agent is used in the decision layer to choose actions that are used in interface adaptation within a constrained design space, altering the existing interface configuration in a controlled way. The execution and governance layer checks every proposed action against accessibility, responsiveness, and safety guardrails, executing only those that meet all constraints and rolling back to a conservative development

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 10, No. 2: 242-259, 2026
DOI: 10.55214/2576-8484.v10i2.12015

configuration if a violation or risk threshold is detected. Practically, the framework facilitates session-level personalization, which is fast and capable of within-session personalization, and longitudinal personalization, which can continuously refine over time with policies specifically designed to reduce disruptive churn at the interface. Combined with these architectural decisions, this ensures that personalization remains inherently human-focused: learning objectives are based on user experience results, and the deployment pipeline enforces controls to keep the user in control, predictable, and trustworthy.

4.2. User Modeling (Preferences, Context, Behavior)

This modeling technique can be used to do actual personalization instead of optimization to one and a generic best user interface. The framework has a tight but concise user representation, which is constantly updated through time. The preferences can be explicit, e.g., language choice, font size, high contrast modes, layout density, or interaction style (guided versus expert), and implicit, e.g., frequent filter usage, regular practice, or avoidance of dense layouts. An example of contextual variables is the device type, viewport size, input modality, network conditions, time of day, and task type, all of which determine the suitability of UI adaptations. Interaction traces generate behavioral cues that are used to identify expertise and interaction friction. The number of backtracks using help-seeking behaviors and long pauses, frequent use of help, and repairing of errors are signs of confusion or high cognitive load, and fast navigation with minimum errors is a sign of proficiency. Task abandonment could be a sign of dissatisfaction or non-alignment of the interface.

This is achieved by updating the user model online based on a combination of long-term characteristics (e.g., accessibility requirements) and short-term situational states (e.g., time pressure) so that the adaptation process can be stable and context-sensitive. It is privacy-aware, and representations of users are stored as aggregated embeddings or temporally summarized statistics instead of interaction logs. It can also be uncertainty-conscious. In cases of low confidence or cold start, the adaptation policy is conservative, restricting the extent and rate of interface modification.

4.3. UI Representation and Design Space Encoding

The creation of individualized UI must have an interface representation that is both machine- and design-viable. Under the proposed structure, every UI setup is coded as a structured object based on a subsystem of design. In this context, multiple forms of such representation exist, such as (i) a hierarchical structure of screens, containers, and components; (ii) a component graph with nodes representing UI components and edges capturing spatial or functional relationships; or (iii) a tokenized sequence of layout and component tokens marked with properties attached to them.

All representations of UIs store semantic attributes (e.g., component type, functional importance, grouping level), spatial attributes (e.g., position, hierarchy, containment), and presentation attributes (e.g., visual density, font size, theme, contrast). The space of all possible interface configurations allowed by the design system and its transformation rules is called the overall design space, ensuring the learning agent cannot produce arbitrary, inconsistent, or non-compliant designs.

The framework specifies a discrete space of meaningful and reversible UI actions to make the action space tractable and the adaptation process interpretable. These involve rearranging groups of components, rescaling elements based on priority, replacing widgets with functionally equivalent components, adjusting density or theme within a predetermined range, and changing the degree of guidance or disclosure. Everything is limited by the template level; it is impossible to remove navigation elements or hide important actions. There are two important benefits to this formal encoding of the UI design space: it makes the design space easier to learn because invalid actions are distinguished in advance, and it maintains usability, brand consistency, and design integrity because all interfaces generated solely consist of accepted design primitives.

4.4. DRL Agent Design

The DRL agent is designed to choose actions for UI adaptation to maximize a long-term, human-centered reward. The framework supports both discrete and continuous action spaces. For actions

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 10, No. 2: 242-259, 2026
DOI: 10.55214/2576-8484.v10i2.12015

involving discrete interface changes, such as repositioning components, value-based approaches like Deep Q-Networks (DQN) or their distributional variants are appropriate. When parameters require continuous adaptation, such as visual density, font scaling, animation strength, or timing, actor-critic methods like Proximal Policy Optimization (PPO) or Soft Actor-Critic (SAC) are more stable and expressive. The agent employs a standard actor-critic architecture, consisting of a policy network that generates action distributions or continuous parameters, and a value network that estimates the expected value of the current state. Both networks operate on a joint state representation, which includes user embeddings, contextual features, UI representations, and recent interaction history. If the UI is modeled as a graph or hierarchical structure, graph neural network encoders or transformer-based token encoders can be used to learn compact UI encodings. In smaller or more constrained design spaces, a structured multilayer perceptron with engineered features may be employed.

The regularization process is done to avoid unstable or excessive interface churn through change-cost penalties and reduced frequency and magnitude of adaptations. The decisions that can be made via adaptation are limited to clearly defined decision points, e.g., between tasks, after screen transition, or task completion events. The agent can also be hierarchical, whereby a high-level policy is used to define what type of adaptation (e.g., layout adjustment, widget replacement, or guidance adjustment) is desired, and a low-level policy is used to define what the specific target and parameterization should be. Such hierarchical decomposition makes the action space simpler and more interpretable without impacting expressive personalization properties.

4.5. Human-Centered Reward Design

The reward structure in this model is clearly anthropocentric and is not confined to performance maximization. The reward function is developed as a multi-objective signal that concurrently considers task success, task efficiency, and user experience. Task performance is quantified using indicators such as task completion, accuracy, and success status, while efficiency is measured by metrics including time on task, number of interaction steps, depth of navigation, and error rate. The results of user experience are included explicitly and implicitly. Explicit measures include post-task tests like satisfaction ratings, System Usability Scale (SUS), or NASA-TLX under controlled experimental conditions. Implicit UX cues are based on behavioral proxies such as task abandonment, undo frequency, repeated backtracking, and extended hesitation.

Reward construction is done with specific consideration to prevent pathological optimization. To illustrate, the decision to optimize only on speed can result in cluttered interfaces, and reducing the number of interaction steps may lead to incentive-focused, manipulative, or misleading designs. To reduce these risks, the reward functionality encompasses penalties for disruptive interface modifications and constraint pressures to encourage stability, predictability, and user confidence. Each component of the rewards is brought to a similar scale, and the weight of the relative components is determined by sensitivity analysis or preference calibration. Critically, the reward functionality is explicitly humanistic. It directly codifies the need to consider comfort, trust, and perceived control, and it penalizes sudden or radical changes in the interface. This design promotes behaviors of learning, which are not only effective in attaining task objectives but also resonate with positive and sustainable user experiences.

4.6. Constraints, Safety Guardrails, and Rollback Strategy

Since the personalization of UI can affect user confidence and cause possible security threats, the framework provides stringent guardrails during decision time and at execution time. These guardrails comprise accessibility checks, responsiveness checks, and usability constraints. The policy rules entail security- and safety-critical workflows. The agent is not allowed to disable safeguards, hide information related to risk, or modify controlled interactions outside of approved templates. Moreover, the system implements change-rate constraints to ensure that oscillations are not too fast or too many changes are made on the interface, maintaining stability.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 10, No. 2: 242-259, 2026
DOI: 10.55214/2576-8484.v10i2.12015

Rollback strategy is activated upon violation of a hard constraint, forecasted risk is above a specified limit, or early signs of interaction indicate that performance is declining. On rollback, the system will restore to the last safe state or a conservative baseline interface, record the event for offline analysis, and may temporarily limit exploration in similar scenarios. This is achieved through mechanisms that make the system safe and reliable during deployment phases, while offering an efficient way to manage uncertainty, non-stationary user behavior, and unexpected interaction patterns in the adaptive interface.

4.7. Training Strategy

The strategy for training an agent to operate safely is to provide an environment for learning, separated by a pre-training phase before going live, so that users do not suffer from any untested behaviors when the agent is performing its tasks. In the pre-training phase of the training strategy, the agent learns how to adapt by studying the interaction logs of simulated users, which enables the agent to learn generalized patterns of adaptation. Once logs of user interactions have been gathered from the live environment, either offline reinforcement learning or imitation learning will be used to establish an adequate personalized policy for the given application. After a personalized policy has been established, onboard coaching using online learning is integrated to take advantage of real-time user feedback, while maintaining a structured process for monitoring the agent's behavior and enabling only conservative exploration of behaviors. In a cold start scenario, a hybrid strategy is adopted, providing a stable base interface with a minimal amount of safe or risk-free adaptations based on explicit user preferences and population-level prior data learned offline. The agent gradually increases the level of personalized adaptation as interaction history becomes available. In the early stages of adaptation, the agent will only introduce coarse-grained adjustments and then transition to fine-grained adaptations. To reduce overfitting due to short-term noise, the framework employs smoothing mechanisms and stability constraints. Learning occurs on two temporal scales. The first scale is global policy training (trained across multiple users) to serve as an overall view of personalization strategies that are generalizable across users. The second scale is user-specific training that gets updated on a per-user basis. Employing this dual-layer training approach, it allows for strong personalization yet retains robustness and generalization across the user group.

4.8. Explainability and User Control

The adaptive interfaces should be human-centered, explainable, and controllable since the user should be able to know and trust what is behind the changes in the interface. The framework has user control mechanisms. Users can reverse or roll back adaptations, freeze certain portions of the interface, set the level of personalization, and rate any given changes as either helpful or not. These controls have a functional and educational purpose. They retain user autonomy and, on top of it, offer monitored signals that can be integrated into the process of personalization. Moreover, a record of all adaptation decisions and constraint evaluations is recorded to facilitate auditing, monitoring, and debugging. This ability is specifically significant in enterprise and regulated settings, where interface modifications should be transparent, traceable, and responsible.

5. Implementation, Evaluation, and User Study Methodology

The process of fully implementing the proposed human-centered DRL personalization framework involves benchmarking the new method against an established baseline and validating its effectiveness with well-controlled user studies. The physical representation of this framework as a modular prototype consists of four interconnected component types.

The UI Rendering Layer produces digital visual representations (i.e., configuration files) for UI components by drawing upon the content available in the approved design system component library (i.e., standardized component and layout templates). Once components and configuration files have been produced, they are loaded into a web and/or mobile front-end development environment (i.e., where users will interact with the components).

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 10, No. 2: 242-259, 2026
DOI: 10.55214/2576-8484.v10i2.12015

User Interaction Logging Layer captures user interaction events such as clicking, tapping, scrolling, dwell time, and navigation depth, synthesizing this data into step-level and macro-level (session) summaries and reports.

The personalization controller contains two major component types: the user model and the DRL Agent. The Personalized Controller determines when to execute adaptations associated with user interaction events based on defined Breakpoints occurring naturally in the workflow. Once these adaptations have occurred, the personalization controller uses the updated configuration files produced to implement them upon completion of the micro-task.

The safety/constraints validation layer verifies that all proposed adaptations comply with prior approved safety/constraint standards, including accessibility, responsiveness, usability, and security. If any action proposed by an adaptation violates any of these standards, the adaptation is denied and processed either as a no-operation or a conservative safe update. If performance has degraded or safety/constraint violations occur, the system will automatically revert to an acceptable configuration previously validated as compliant with all prior approved safety/constraint standards.

5.1. Environment Simulator

An environment simulator provides a consistent user-UI interaction loop for scalable training and repeatable testing. In simulation mode, users provide a UI configuration. From that, the environment will return simulated responses based on a synthetic user model of real statistics generated from response data. The next state and reward for the episode are also from this loop. In real-user mode, the user interaction log may be live or recorded. During this mode, updates can be made offline, while the user will still be receiving the "guard rails" of the environment.

5.2. State and UI Encoding

State encoding in the proposed framework is compact, privacy-preserving, and personalized. Each state representation integrates a user embedding that captures inferred preferences and expertise, contextual features such as device type, viewport size, and task category, a structured representation of the current UI configuration, and short-term behavioral summaries (e.g., error occurrences, backtracking frequency, hesitation indicators, and progress signals). UI configurations, modeled as hierarchies, layouts, or graphs, encode component types, semantic roles, visual density, font scaling, theme attributes, and adjacency relationships. Compact UI embeddings are derived either through a lightweight transformer operating on tokenized UI representations or through a graph-based encoder when preserving structural dependencies is critical.

5.3. Action Space & Features

The actions are described as interpretable and reversible interface changes within a narrow and specified design space. Examples include the re-arrangement of collections of interface items, the elevation of commonly used functionality, aggregating between approved alternatives of a widget, resizing visual density or font size within safe ranges, and activating or deactivating guidance or aid systems. Such actions are strictly restricted so that adaptations are understandable, controllable, and obey usability and safety restrictions.

The framework uses a two-stage action selection strategy to avoid the combinatorial explosion of the action space. At the initial step, the agent chooses an area of adaptation (e.g., layout change, widget replacement, or guidance change). The second stage involves the agent making the choice of the specific target components or parameters to be changed, subject to the selection of the category that was chosen, as specified in the relevant templates. This hierarchy makes the actions less complex and more interpretable. The feature extraction also facilitates proper decision-making by normalizing interaction signals over time, aggregating raw event logs into constant and defined metrics, and extracting higher-level user experience (UX) proxy measurements. Such proxies can include estimating a "confusion

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 10, No. 2: 242-259, 2026
DOI: 10.55214/2576-8484.v10i2.12015

probability" in terms of repetitive behavioral patterns, e.g., frequent backtracking, repeated error correction, or long hesitation times within a specified time window.

5.4. Training Setup

The strategy for training DRL agents follows a safety-first approach. Training of the DRL agent will be accomplished using stable training algorithms such as actor-critic algorithms (e.g., PPO) to accomplish constrained updates or value-based algorithms when the action set is strictly discrete. The hyperparameters used for tuning will be: learning rate, discount factor, entropy regularization, batch size, update frequency, and constraint penalty weights. The weights assigned to the reward functions during the tuning process will be determined through sensitivity analysis to ensure that task performance is balanced against user comfort, trust, perceived control, disruption penalties, and efficiency.

5.5. Pretraining and Cold Start

The initial part is usually training in the offline pretraining stage, where the DRL agent is trained with recorded interaction data and/or simulated users to learn a starting, useful policy. This step is succeeded by a cautious web-based education phase, acquiring actual user comments within narrow exploration and rigid safety protocols. In cold-start situations, the agent works on a stable baseline interface with minimal and safe personalization according to explicit user preferences, including language, font size, and accessibility modes, and population-level priors. As more proof is gathered and trust in the user model is built, the framework moves toward increasingly powering through personalization without losing stability or user trust.

5.6. Runtime and Fallbacks

The primary evaluation criteria used in core deployment requirements are runtime constraints. The optimization of policy inference is based on the constraints of interactive latency, accessibility rules, layout breakpoints, and prohibited interactions, validated with deterministic checks whose costs are linear in the number of UI components. To further minimize latency, the system stores UI embeddings of interface settings that have already been displayed to the user. Decisions to be adapted can only be made when the user can tolerate short delays naturally, such as when changing the screen or at the end of a task. If policy inference takes longer than the acceptable latency limits or cannot arrive at a sufficiently certain decision, the system will safely revert to the last validated configuration or use a no-op update. This design will be responsive, robust, and will provide a smooth user experience under real-world deployment conditions.

5.7. Evaluation Setup, Baselines, and Metrics

The assessment was performed based on the combination of real-life interaction records, artificial records of interaction, and known task situations. The following strategies have been contemplated as multiple baselines that could be compared: (1) a non-personalized fixed interface; (2) rule-based strategies of personalization (e.g., frequency-based shortcuts and heuristic-driven layouts); (3) a supervised model of preference prediction; (4) one-shot adaptations through contextual bandits; (5) an ablated version of the proposed framework, including the exclusion of a major different component, (a) the UX reward term, (b) the disruption penalty, or (c) the safety.

The metrics of evaluation were classified into three major categories. Time on task, number of interaction steps, and error rate were also considered as task performance metrics. The user experience (UX) results were measured with the help of standardized tools, including the System Usability Scale (SUS) [22] and NASA-TLX [23], as well as satisfaction, trust, and perceived control measures, and further behavioral proxies, which include abandonment and undo actions. The metrics used to measure safety and system stability included the magnitude of interface change between consecutive

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 10, No. 2: 242-259, 2026
DOI: 10.55214/2576-8484.v10i2.12015

configurations, adaptation frequency, rollback rate, action rejection rate, attempted constraint violations, and the frequency of accessibility regressions.

5.8. Ablation, Robustness, and Statistical Analysis

The evaluation of key component contributions is conducted through ablation and similar experiments using HRM (Machine Learning) implementation. The robustness test assesses the impact of telemetry noise, telemetry loss, new tasks, hardware or software environment changes, or battery failure issues. Confidence intervals, statistical testing with Bonferroni and Sidak method [24], effect sizes; not met, exploratory statistical techniques like Mann-Whitney U test [25] or bootstrapping [26]. Training dispersion is also reported as the average and the dispersion of RL learning curves obtained with various random seeds.

The study methods are either a counterbalanced within-subject design or a between-subject design, depending on the possibility of carry-over effects, and disclose participant workflows either through a baseline UI (the default setting), a heuristic personalized UI (via rules), or an HC-DRL personalized UI (configured individually). Each of the UI conditions is then measured in both usability measures (SUS) and workload measures (NASA-TLX), as well as perceived metrics of trust, control, transparency, and reliability. Time, number of steps, errors, and other behaviors or experiences are all combined with the subjective feedback into a single dataset. The qualitative feedback is gathered through interviews or open-ended questions and is themed to analyze the data. Other ethical considerations involve informed consent of all participants, the right to withdraw at any point, minimal required data collection, preference to use aggregated telemetry data as opposed to content of records, and security of study data.

6. Results

The empirical findings of the proposed HC-DRL model for creating tailored UI are presented, organized to reflect the most important assessment goals. The validation of RL convergence stability, positive improvements over baselines in task-solving and UX, personalization stability, and adherence to constraints are demonstrated. Large gains are attributed to the proposed human-friendly elements, reward shaping, disruption control, and guardrails.

6.1. Convergence and Learning Curves of RL.

The HC-DRL agent demonstrates consistent training dynamics. In the first episodes, the policy goes through the policy tightness space of UI, which leads to a greater range of episodic returns variation and more frequent occurrence of an attempt at invalid or disruptive behavior. The episodic return increases continuously until it flattens as training advances, suggesting that the agent is no longer interested in leveraging short-term variations to maximize utility but instead maximizes utility in the long term. The tendency to converge is indicated by the plateau of the moving average of returns, smaller variance among random seeds, and a steady decrease in the number of constraint violations attempted, showing that the policy effectively implements feasibility and safety constraints (Table 1). The major convergence indicators are outlined in Table 8.1, such as the speed at which the agent achieves a large share of its optimal performance and the visible decrease in the number of unsafe action proposals.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 10, No. 2: 242-259, 2026
DOI: 10.55214/2576-8484.v10i2.12015

Table 1.
RL learning behaviour and convergence statistics of the HC-DRL model.

Metric (HC-DRL)	Value
Initial episodic return (first 100 episodes)	0.78 ± 0.19
Final episodic return (last 100 episodes)	2.31 ± 0.12
Episodes to reach 90% of the final return	1.180 ± 160
Episodes to reach 95% of the final return	1.650 ± 220
Variance of return (early → late)	0.31 → 0.08
Attempted constraint-violating actions (early → late)	6.4% → 1.6%

6.2. Task Performance Results vs Baselines

The performance of the tasks is measured by the completion rate, time-on-task, number of interaction steps, and the error rate. The proposed approach has better overall task results than all the other baseline approaches. Compared to the non-executive UI status, HC-DRL improves completion rates and minimizes the time spent on the task and the number of errors, which indicates that individual decision-making generates quantifiable efficiency changes and not only aesthetic modifications. As compared to heuristic personalization and contextual bandit, the suggested strategy benefits from long-horizon optimization: it will identify the right time to change, as well as the scale of UI change to maximize future interaction performance, and not only prioritize short-term click-level performance. Table 2 shows a detailed comparison of the task-performance indicators, which demonstrates a steady increase in the main indicators.

Table 2.
Task performance comparison across personalization methods.

Method	Task completion (%) ↑	Time-on-task (s) ↓	Steps/task ↓	Errors/task ↓
Static UI (no personalization)	86.2 ± 2.1	94.6 ± 8.3	14.8 ± 1.6	1.42 ± 0.22
Heuristic personalization (rules)	89.4 ± 1.8	87.2 ± 7.5	13.6 ± 1.4	1.23 ± 0.18
Contextual bandit (one-step)	90.8 ± 1.6	83.4 ± 6.9	13.1 ± 1.3	1.18 ± 0.16
Supervised preference predictor	90.1 ± 1.7	85.1 ± 7.2	13.3 ± 1.4	1.20 ± 0.17
Proposed HC-DRL (ours)	93.7 ± 1.2	76.5 ± 6.1	12.1 ± 1.1	0.92 ± 0.13

6.3. UX Outcomes and User-Study Results

In addition to task efficiency, the structure is created in such a way that it improves the user experience and their perceived autonomy. Empirical evidence from a user study shows that HC-DRL scores the highest on the System Usability Scale (SUS) and the lowest on workloads according to NASA-TLX, along with higher satisfaction and perceived control/trust. These findings indicate that the reward schema effectively balances task performance with human concerns, including comfort and perceived predictability. Qualitative observations are typically consistent with the quantitative data: users reported feeling positive about personalization that did not significantly restructure the interface, especially when changes were subtle and conformed to existing patterns of use. The summary of the user experience outcomes of the methods considered is presented in Table 3.

Table 3.
Comparison of user experience results for different personalization approaches.

Method	SUS ↑	NASA-TLX ↓	Satisfaction (1–5) ↑	Trust/Control (1–5) ↑
Static UI	68.1 ± 6.5	52.4 ± 7.1	3.38 ± 0.55	3.42 ± 0.54
Heuristic personalization	71.6 ± 5.8	48.9 ± 6.4	3.55 ± 0.51	3.55 ± 0.51
Contextual bandit	72.8 ± 5.2	47.6 ± 6.0	3.60 ± 0.50	3.61 ± 0.49
Supervised predictor	73.4 ± 5.5	46.8 ± 5.7	3.66 ± 0.48	3.66 ± 0.48
Proposed HC-DRL (ours)	80.2 ± 4.9	38.7 ± 5.2	4.08 ± 0.41	4.12 ± 0.39

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 10, No. 2: 242-259, 2026
DOI: 10.55214/2576-8484.v10i2.12015

6.4. Stability/Disruption and Constraint Violations

The main risk associated with adaptive user interfaces is instability, where frequent or significant changes can undermine predictability and user trust. The proposed framework addresses this risk through disruption penalties, state-dependent evaluation of feasible actions, and the inclusion of hard guardrails that may trigger rollbacks. Stability is assessed based on how much layout and behavior change across sessions, with a disruption measure capturing the scale of these changes. Safety parameters, such as the frequency of guardrail rejections and rollback events, are also monitored. The HCDRL approach presented in this study achieves a lower level of disruption compared to learning baselines while maintaining high task and user experience performance. It is important to note that constraint violations in the deployed system are considered a null hypothesis, as invalid actions are not permitted before deployment. Constraint violations tend to decrease as the policy converges during early training epochs. The stability and safety metrics are detailed in Table 4.

Table 4.
Stability and safety constraint compliance results.

Method	UI changes/session ↓	Disruption score ↓	Guardrail rejects (%) ↓	Rollbacks (%) ↓	Rendered violations (%) ↓
Static UI	0.00 ± 0.00	0.00 ± 0.00	0.0	0.0	0.0
Heuristic personalization	0.9 ± 0.3	0.21 ± 0.07	1.8 ± 0.6	0.7 ± 0.3	0.0
Contextual bandit	1.4 ± 0.5	0.29 ± 0.09	2.6 ± 0.9	1.2 ± 0.5	0.0
Supervised predictor	1.1 ± 0.4	0.25 ± 0.08	2.1 ± 0.7	1.0 ± 0.4	0.0
Proposed HC-DRL (ours)	0.8 ± 0.3	0.17 ± 0.06	1.3 ± 0.5	0.6 ± 0.2	0.0

6.5. Ablation and Robustness Results

The experiments of ablation show that the improvements are related to the human-centered components and not the DRL component alone. The elimination of UX reward terms tends to hasten speed-based behavior, but at the same time reduces usability and adds workload, thus highlighting the need to explicitly optimize human-centered goals. Removal of disruption penalties leads to increased oscillatory adaptations, often known as UI churn, and hence increases rollback frequency and degrades the feeling of control over the user. The deactivation of guardrails returns artificially inflated simulated returns but provides unsafe proposals, making the system not suitable for use. Stress tests of the algorithm in noisy telemetry, signal dropout, and drift of preference conditions indicate that the algorithm's performance decays gracefully due to its conservative fallback mechanism and user model, which reduce the short-term effects of noise. The summary of these results is in Table 5 in the form of deltas against the full HC-DRL model.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 10, No. 2: 242-259, 2026
DOI: 10.55214/2576-8484.v10i2.12015

Table 5.
Ablation and robustness analysis results.

Variant / Test	Δ Completion (pp)	Δ Time (s)	Δ Errors	Δ SUS	Δ TLX	Δ Disruption	Δ Rollbacks (pp)	Key observation
Ours without UX reward terms	−2.4	−4.1	−0.03	−7.8	+6.9	+0.09	+0.7	Faster but worse UX/trust
Ours without a disruption penalty	+0.6	−2.8	+0.01	−5.1	+3.8	+0.22	+1.6	UI churn and instability
Ours without guardrails (analysis only)	+1.2	−5.6	−0.02	−9.4	+8.1	+0.31	+3.9	Unsafe; many invalid actions
Noisy telemetry (+20% noise)	−1.1	+2.9	+0.07	−2.6	+2.1	+0.05	+0.4	Graceful degradation
Missing signals (remove dwell/help)	−1.6	+3.7	+0.09	−3.3	+2.7	+0.06	+0.5	More conservative behavior
Preference drift mid-session	−0.9	+2.1	+0.05	−2.0	+1.6	+0.04	+0.3	Recovery depends on update speed

7. Discussion

This work shows that the problem of personalized UI generation can be conceptualized as a long-horizon, human-oriented decision-making process and addressed by deep reinforcement learning with a stringent design budget. The overall findings indicate that maximization of task efficiency is not ideal in adaptive interfaces in real-life situations and that explicit specification of user experience, stability, and safety requirements is also necessary in achieving personalization that can be enjoyed and approved by the user in the long run. The following discussion decodes the empirical results, explains the trade-offs that the experiments have discovered, and presents the implications and practical implications related to the implementation of such systems.

One of the main inferences is that the suggested HCDRL framework positively affects task performance and, at the same time, makes the user experience better, which proves that personalization can be beneficial when it is supported with human-centered goals. Improved completion rate, time-on-task, and reduction of errors suggest that the agent acquires significant adaptations to friction reduction and not just aesthetic arrangement change. At the same time, the high level of usability and trust/control is also indicative of the reward design being effective in focusing on the priority of comfort and perceived autonomy. This interpretation is further supported by the learning dynamics. The more training is done, the smaller the attempted constraint violations and rollbacks are, and this suggests that the policy learns to internalize the feasibility boundaries and can be more conservative when making suggestions. This trend is noteworthy as it implies that this agent is not just gaming the reward but learning to act in a manner that is effective in the safe design space, where successful personalization will have to work.

There is also a steady trade-off between efficiency and stability/comfort, which is seen in the experiments. Specialized interfaces are not always better as viewed by the user. Ablations and baseline comparisons demonstrate that elimination of UX-biased reward words or disruption fines may be able to provide short-term efficiency but lead to more instability and reduce felt usability and control. The above observation implies that adaptive UI policies have to be regularized with respect to disruptive behavior and that they need to include explicit feedback pertaining to comfort, either directly by user measurements or indirectly by interaction proxies. The identified trade-off can be idealized as an optimization tension: when reconfiguring the UI assertively to reduce time or steps, a policy can sacrifice predictability and increase cognitive load, but a policy that does not push change can fail to personalize significantly. The proposed solution to this dilemma is to charge change-cost conditions, restrict the adaptation rate, and use hard guardrails. Practically, the least objectionable kind of personalization is often incremental and behavior-compatible, with small changes like prioritizing

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 10, No. 2: 242-259, 2026
DOI: 10.55214/2576-8484.v10i2.12015

commonly used actions, reducing density within recommended thresholds, or allowing context suggestions. These are seen as supportive, whereas much larger rearrangements and swings in the opposite direction can destroy usability, even with increased performance measures.

These results have direct design implications for adaptive UIs. First, the concept of human-centered needs must be operationalized, not merely proclaimed: UX results and perceived control should be hard-coded, and reward functions validated through user experiments, rather than viewed as secondary effects of performance increases. Second, a design system and usability rules should limit the design space so that the actions of an agent are always design-valid; this not only simplifies reinforcement learning but also enhances reliability. Third, adaptation must be event-based and rate-based: implementing changes only at natural boundaries (e.g., between tasks or screens) improves user acceptance and minimizes disruption. Fourth, personalization must be uncertainty-conscious, especially in cold-start situations: when user model confidence is low, the policy should adopt a conservative, reversible approach. Lastly, user control is an essential aspect of the system. Features like undo, lock regions, explicit feedback, and adjustable personalization mechanisms not only increase user trust but also provide valuable learning cues to stabilize the policy and prevent negative feedback loops of personalization.

Even though encouraging results have been found, there are a number of limitations and threats to validity. The first is the issue of ecological validity. Even carefully designed laboratories can be unsuccessful in the real world at modeling the complexity, the drift of preferences over time, and the multi-tasking workflow. Second, offline logs and user simulation may be biased. Synthetic users might not be representative of the variety of real interaction strategies, and recorded data can constrain the structure of the reference UI, limiting exploration of alternative designs. Third, UX proxies have measurement validity problems: dwell time or backtracking can be the result of confusion, but can also be due to careful reading or intentional decision-making; the judgment about latent states (frustration, workload) is not yet perfected. Fourth, generalization can be limited by the chosen design space and application setting: a policy learned in one setting within a UI ecosystem might not be directly applicable to a different environment without retraining or changing option encodings and reward weights. Fifth, the RL optimization process also introduces variation and sensitivity to hyperparameters, reward scaling, and exploration schedules, although convergence can be discerned empirically. Its stability in general conditions should be tested more broadly. These shortcomings indicate that future research needs to be based on longitudinal field research, more multi-domain data, and uncertain user state generalization, as well as automated device and UI family testing.

Privacy, latency, and transparency are critical in terms of deployment. Privacy also requires that data collection should be kept to a minimum, and that only the information necessary to personalize the business is stored, in the form of aggregation or embedding (not in the form of actual interaction contents). Opt-in consent, distinct data retention policies, and deletion of the history of personalization should be supported by user identification and long-term personalization. Latency budgets demand state creation, policy derivation, and guardrail checks are carried out with interactive budgets; real-world systems via caching embeddings in the UI, restrict decision points, and offer deterministic fallback behavior whenever inference is slow or indeterminate. Trust requires transparency and accountability: users must know that personalization is taking place, must be able to understand why significant changes have occurred, and must have easy controls to decrease or turn off personalization. Explanations about observable behavior that are lightweight can be more acceptable, but should not make claims about assumptions regarding internal states. Further safe operation requires sustained monitoring: record-keeping of adaptation decisions, guardrail triggers, and rollback events can enable auditing and quick refinement, and A/B testing frameworks can be used to justify incremental policy changes by ensuring they do not provoke massive instability.

This discussion has reiterated the fact that successful personalized UI generation is not strictly a reinforcement learning issue but a human-centered systems issue. The suggested framework also works well in its performance since it combines RL with limited design space, stability-constrained objectives,

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 10, No. 2: 242-259, 2026
DOI: 10.55214/2576-8484.v10i2.12015

and user-friendly control systems. The findings and ablations all testify to the fact that comfort, predictability, and safety have to be maximized expressly to ensure personalization not just in an efficient manner, but also in a manner that is not only trusted but also sustainable in practical deployments.

8. Conclusion

The paper introduces a Human-Centered Deep Reinforcement Learning (HC-DRL) model for generating personalized user interfaces, addressing the mismatch between increasingly complex digital systems and actual user needs. The framework formulates personalization as a constrained sequential decision-making problem, enabling the learning of policies that optimize interaction quality over the long term rather than just short-term performance. It integrates an evolving user model, a structured UI representation conformant to design systems, and a DRL agent whose training is guided by a human-friendly reward system. This reward balances success and efficiency in tasks with factors such as satisfaction, cognitive comfort, trust, perceived control, and punitive measures against disruptive modifications. To ensure safe real-world behavior, the framework incorporates constraint-based guardrails and rollback mechanisms that preserve accessibility, usability, responsiveness, and safety, reducing the risk that aggressive adaptations will undermine predictability and user confidence. The contribution extends beyond applying DRL to personalization, establishing a direct link between learning goals and human experience, and proposing governance mechanisms to support implementation. The formulation accommodates both MDP and POMDP models, recognizing that latent user causes, such as intent, frustration, and workload, can only be partially observed and estimated through interaction signals. Future directions include increased focus on longitudinal field experiments, improved modeling of preference changes, privacy-conscious offline-to-online training, and enhanced transparency and user management systems to foster greater trust and adoption.

Transparency:

The author confirms that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.

References

[1] M. W. Iqbal, N. A. Ch, S. K. Shahzad, M. R. Naqvi, B. A. Khan, and Z. Ali, "User context ontology for adaptive mobile-phone interfaces," *IEEE Access*, vol. 9, pp. 96751-96762, 2021. https://doi.org/10.1109/ACCESS.2021.3095300

[2] K. Upreti, P. Kumari, U. Shankar, G. Radhakrishnan, S. Udhaya, and K. Malik, *Human–computer interaction for cognitive, emotional and learning well-being. Intelligent Systems for Neurocognition and Human-Robot-Computer Interaction.* Cambridge, MA, USA: Elsevier, 2026.

[3] A. Khamaj and A. M. Ali, "Adapting user experience with reinforcement learning: Personalizing interfaces based on user behavior analysis in real-time," *Alexandria Engineering Journal*, vol. 95, pp. 164-173, 2024. https://doi.org/10.1016/j.aej.2024.03.045

[4] M. Murtaza, Y. Ahmed, J. A. Shamsi, F. Sherwani, and M. Usman, "AI-based personalized e-learning systems: Issues, challenges, and solutions," *IEEE access*, vol. 10, pp. 81323-81342, 2022. https://doi.org/10.1109/ACCESS.2022.3193938

[5] T. M. Singh, C. K. K. Reddy, B. R. Murthy, A. Nag, and S. Doss, *AI and education: Bridging the gap to personalized, efficient, and accessible learning. In M. Ouaissa, M. Ouaissa, H. Lamaazi, M. El Hamlaoui, & C. K. Reddy (Eds.), Internet of Behavior-Based Computational Intelligence for Smart Education Systems.* Hershey, PA, USA: IGI Global, 2025.

[6] H. Farhood, M. Nyden, A. Beheshti, and S. Muller, "Artificial intelligence-based personalised learning in education: A systematic literature review," *Discover Artificial Intelligence*, vol. 5, p. 331, 2025. https://doi.org/10.1007/s44163-025-00598-x

[7] M. İ. Berkman, *Introduction: Experience in extended realities. In Ö. Cordan, M. İ. Berkman, G. Çatak, & D. Arslan Dinçay (Eds.), Extended realities, virtual environment, and interactive experiences*. Boca Raton, FL, USA: CRC Press/Taylor & Francis Group, 2025.

[8] J. H. Nderitu, "Mental state adaptive interfaces as a remedy to the issue of long-term continuous human machine interaction," *Journal of Robotics Spectrum*, vol. 1, pp. 078-089, 2023.

[9] D. Gaspar-Figueiredo, M. Fernández-Diego, R. Nuredini, S. Abrahão, and E. Insfrán, "Reinforcement learning-based framework for the intelligent adaptation of user interfaces," in *Companion Proceedings of the 16th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, 2024.

[10] N. A. Hafez, M. S. Hassan, and T. Landolsi, "Reinforcement learning-based rate adaptation in dynamic video streaming," *Telecommunication Systems*, vol. 83, no. 4, pp. 395-407, 2023. https://doi.org/10.1007/s11235-023-01031-3

[11] P. Lv *et al.*, "User-guided personalized image aesthetic assessment based on deep reinforcement learning," *IEEE Transactions on Multimedia*, vol. 25, pp. 736-749, 2021. https://doi.org/10.1109/TMM.2021.3130752

[12] H. Du *et al.*, "Diffusion-based reinforcement learning for edge-enabled AI-generated content services," *IEEE Transactions on Mobile Computing*, vol. 23, no. 9, pp. 8902-8918, 2024. https://doi.org/10.1109/TMC.2024.3356178

[13] X. Xu, G. Feng, S. Qin, Y. Liu, and Y. Sun, "Joint multi-UAV deployment and resource allocation based on personalized federated deep reinforcement learning," in *ICC 2023-IEEE International Conference on Communications (pp. 5677-5682). IEEE*, 2023.

[14] X. Xu, G. Feng, S. Qin, Y. Liu, and Y. Sun, "Joint UAV deployment and resource allocation: A personalized federated deep reinforcement learning approach," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 3, pp. 4005-4018, 2023. https://doi.org/10.1109/TVT.2023.3328609

[15] Q. Wu, W. Wang, P. Fan, Q. Fan, H. Zhu, and K. B. Letaief, "Cooperative edge caching based on elastic federated and multi-agent deep reinforcement learning in next-generation networks," *IEEE Transactions on Network and Service Management*, vol. 21, no. 4, pp. 4179-4196, 2024. https://doi.org/10.1109/TNSM.2024.3403842

[16] X. Li, R. Zhou, Z. C. Lipton, and L. Leqi, "Personalized language modeling from personalized human feedback," *arXiv preprint arXiv:2402.05133*, 2024.

[17] Q. Sun *et al.*, "Symbioticrag: Enhancing document intelligence through human-llm symbiotic collaboration," *arXiv preprint arXiv:2505.02418*, 2025. https://doi.org/10.48550/arXiv.2505.02418

[18] G. Diao, C. Li, Q. Liu, and Z. Liu, "Empirical study on the application of deep learning in user behavior prediction and personalized recommendation in e-commerce," *Journal of Organizational and End User Computing*, vol. 37, no. 1, pp. 1-36, 2025. https://doi.org/10.4018/JOEUC.383512

[19] F. Yang and Z. Wang, "An intelligent taekwondo coaching system based on augmented reality technology with real-time feedback mechanisms," *Scientific Reports*, vol. 15, p. 40832, 2025. https://doi.org/10.1038/s41598-025-24608-1

[20] Y. Feng and X. Jiang, "SUMFORU: An LLM-based review summarization framework for personalized purchase decision support," *arXiv preprint arXiv:2512.11755*, 2025. https://doi.org/10.48550/arXiv.2512.11755

[21] H. Kurniawati, "Partially observable markov decision processes and robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, no. 1, pp. 253-277, 2022. https://doi.org/10.1146/annurev-control-042920-092451

[22] P. Vlachogianni and N. Tselios, "Perceived usability evaluation of educational technology using the System Usability Scale (SUS): A systematic review," *Journal of Research on Technology in Education*, vol. 54, no. 3, pp. 392-409, 2022. https://doi.org/10.1080/15391523.2020.1867938

[23] K. Virtanen, H. Mansikka, H. Kontio, and D. Harris, "Weight watchers: NASA-TLX weights revisited," *TheoreTical Issues in Ergonomics Science*, Vol. 23, no. 6, pp. 725-748, 2022. https://doi.org/10.1080/1463922X.2021.2000667

[24] V. Taweesapaya, A. Thongteeraparp, W. Wanishsakpong, P. Sudsila, and A. Volodin, "Fuzzy method for multiple hypotheses testing procedure," *Lobachevskii Journal of Mathematics*, vol. 45, no. 9, pp. 4387-4393, 2024. https://doi.org/10.1134/S1995080224605435

[25] R. Wall Emerson, "Mann-Whitney U test and t-test," *Journal of Visual Impairment & Blindness*, vol. 117, no. 1, pp. 99-100, 2023. https://doi.org/10.1177/0145482X221150592

[26] T. Zahel, "A novel bootstrapping test for analytical biosimilarity," *The AAPS Journal*, vol. 24, no. 6, p. 112, 2022. https://doi.org/10.1208/s12248-022-00749-3