

УДК 004.891

Кравцова Екатерина Юрьевна, аспирант, 1 курс, СТАНКИН – Московский государственный технологический университет (МГТУ «СТАНКИН»), 127055, Россия, г. Москва, Вадковский переулок, д. 3-А, Институт информационных технологий, Россия, г. Москва

МОДЕЛЬ АДАПТИВНОГО ИНТЕРФЕЙСА ВЕБ-ПРИЛОЖЕНИЯ НА ОСНОВЕ ФУНКЦИИ ВОЗНАГРАЖДЕНИЯ С УЧЕТОМ ПОЛЬЗОВАТЕЛЬСКИХ ПРЕДПОЧТЕНИЙ

Аннотация

В статье рассматривается проблема интеллектуальной адаптации пользовательского интерфейса веб-приложения. Обосновывается необходимость перехода от статичных интерфейсных решений и заранее заданных правил к модели, в которой адаптация представляется как последовательный процесс принятия решений. В качестве формальной основы используется марковский процесс принятия решений, где состояние описывает конфигурацию интерфейса и пользовательский контекст, действие соответствует изменению UI, а reward (вознаграждение) отражает успешность адаптации. Особое внимание уделяется балансу *generality* и *individuality*, позволяющему сочетать общие закономерности пользовательского опыта с индивидуальными предпочтениями конкретного пользователя. Предлагаемая модель рассматривает адаптивный интерфейс как замкнутый интеллектуальный контур, в котором RL-агент выбирает адаптации, получает оценку результата и уточняет дальнейшую стратегию поведения.

Annotation

The paper addresses the problem of intelligent adaptation of user interfaces in web applications. The necessity of transitioning from static interface solutions and predefined rules to a model in which adaptation is represented as a sequential decision-making process is justified. A Markov decision process is employed as the formal foundation, where the state describes the interface configuration and user context, the action corresponds to UI changes, and the reward reflects the success of adaptation. Special attention is paid to the balance between generality and individuality, allowing the combination of general patterns of user experience with individual preferences of specific users. The proposed model treats the adaptive interface as a closed intelligent loop, in which an RL-agent selects adaptations, receives feedback on results, and refines further behavioral strategy.

Ключевые слова: адаптивный интерфейс, MDP, RL, пользовательский опыт, интеллектуальная адаптация.

Keywords: adaptive interface, MDP, RL, user experience, intelligent adaptation

Введение

Развитие веб-приложений приводит к усложнению пользовательских сценариев. Пользователь все чаще взаимодействует не с отдельной страницей, а с многоэтапной цифровой средой, где требуется искать информацию, сравнивать варианты, возвращаться к предыдущим действиям, уточнять параметры и принимать решения. В таких условиях статичный интерфейс не всегда способен одинаково хорошо поддерживать разных пользователей.

Адаптивный интерфейс позволяет изменять структуру, визуальное оформление и поведение веб-приложения в зависимости от условий взаимодействия. Он может менять компоновку, размер шрифта, плотность информации, тему оформления, подсказки и навигационные элементы. Однако сама возможность изменения UI не решает основную задачу. Важнее

определить, какая адаптация нужна, в какой момент она должна быть выполнена и как оценить ее результат.

Традиционный подход часто основан на правилах. Такие правила могут учитывать устройство, размер экрана, частоту использования функций или заранее заданные пользовательские настройки. Но правила плохо работают там, где пользовательское поведение меняется во времени. Один и тот же пользователь может по-разному взаимодействовать с интерфейсом в зависимости от задачи, контекста и уровня опыта. В такой постановке веб-приложение рассматривается как среда, а интеллектуальный модуль адаптации пользовательского интерфейса выполняет роль управляющего компонента, который выбирает допустимые изменения UI и получает оценку их успешности через функцию вознаграждения

Цель статьи заключается в разработке концептуальной модели адаптивного интерфейса веб-приложения на основе MDP, RL и функции вознаграждения, учитывающей баланс между *generality* и *individuality*.

1. Формализация адаптации UI через MDP

Адаптация интерфейса не должна рассматриваться как единичное изменение внешнего вида страницы. Веб-приложение работает в динамической среде, где пользователь выполняет последовательность действий, а каждое изменение интерфейса влияет на дальнейшее взаимодействие. Если система увеличивает размер шрифта, это может улучшить читаемость, но уменьшить объем информации на экране. Интерфейс скрывает часть второстепенных данных, что снижает перегрузку, но одновременно затрудняет принятие решений. Реализацию перестановки блоков под предпочтения пользователя, на первый взгляд, можно считать оптимальным решением, но частые перестройки каркаса приложения способны нарушить привычную логику работы.

Поэтому адаптация UI требует оценки последствий. Нельзя ограничиться вопросом, может ли система изменить элемент интерфейса.

Важнее определить, приведет ли изменение к улучшению пользовательского опыта. Именно поэтому адаптивный интерфейс целесообразно рассматривать как последовательную задачу, где система выбирает действие, наблюдает результат и уточняет дальнейшую стратегию. Реактивная адаптация выполняет заранее заданное правило. Интеллектуальная адаптация учитывает прошлый опыт, состояние интерфейса, пользовательский контекст и ожидаемый результат.

Марковский процесс принятия решений (Markov Decision Process) используется для описания задач, где агент выбирает действия в изменяющейся среде и получает вознаграждение за результат. В задаче адаптации интерфейса MDP позволяет связать состояние веб-приложения, возможные действия адаптации и оценку успешности изменений.

В рамках предлагаемой модели MDP включает несколько основных элементов:

Элемент MDP	Интерпретация для адаптивного интерфейса
State	текущее состояние UI, пользовательский контекст, параметры задачи
Action	изменение интерфейса или сохранение текущей конфигурации
Reward	оценка успешности адаптации
Policy	стратегия выбора интерфейсных изменений
Agent	интеллектуальный модуль адаптации интерфейса веб-приложения

Состояние отражает все параметры, которые могут повлиять на выбор адаптации. К ним относятся компоновка интерфейса, тема оформления, размер шрифта, объем отображаемой информации, активные подсказки, тип устройства, история действий, пользовательские предпочтения и признаки

затруднения. В исходном RL-фреймворке состояние может включать информацию о пользователе, платформе, среде и параметрах дизайна UI.

Действия соответствуют возможным адаптациям интерфейса. Это может быть изменение цветовой схемы, перестановка элементов, изменение размера шрифта, изменение контента или навигационного потока. Важно, что действие может включать и вариант сохранения текущего состояния, поскольку не всякая ситуация требует активной перестройки UI. Среди действий также выделяется `operate`, позволяющее агенту не менять интерфейс, когда текущая конфигурация является оптимальной.

Reward показывает, насколько успешной оказалась адаптация. Без функции вознаграждения RL-агент не сможет отличить полезное изменение от нерелевантного. Поэтому reward становится центральным элементом модели.

2. RL-агент как интеллектуальный модуль адаптации интерфейса

Обучение с подкреплением позволяет системе постепенно уточнять стратегию адаптации. RL-агент получает состояние интерфейса, выбирает действие, получает reward и на основе результата обновляет свою политику. В адаптивном веб-приложении RL-агент выполняет роль интеллектуального адаптера. Он не создает интерфейс заново, а работает внутри набора допустимых адаптаций. Базовая структура задается проектировщиком, а агент выбирает, какое изменение лучше подходит для текущего состояния, что позволяет учитывать не только текущую ситуацию, но и последствия прошлых адаптаций.

RL-агент может работать с разными типами адаптаций. Например, может выбирать между списочным и сеточным представлением, светлой и темной темой, малым и крупным размером шрифта, полным и сокращенным отображением информации. Такие параметры рассматриваются как часть пространства состояний и действий: `layout`, `theme`, `font size` и `information display`.

Для веб-приложения это особенно важно. Пользовательские сценарии часто отличаются по целям и условиям. Один пользователь работает с большим экраном, другой — с мобильного устройства. Один нуждается в подробных пояснениях, другой предпочитает компактный интерфейс. RL-агент позволяет учитывать такие различия не только через заранее заданные правила, но и через накопленный опыт адаптации.

3. Reward как механизм оценки успешности адаптации

Вознаграждение (reward) является основой обучения RL-агента. Он показывает, насколько выбранное действие оказалось полезным для системы и пользователя. В контексте интерфейса reward нельзя сводить только к одному показателю. Успешность адаптации может выражаться в разных формах: пользователь быстрее завершил задачу, сделал меньше ошибок, не вернулся к прежнему экрану, выбрал предложенный вариант, сохранил вовлеченность или явно подтвердил удобство изменения. Функция вознаграждения должна учитывать специфику пользовательского опыта. Если reward будет задан слишком грубо, агент может обучиться формально правильным, но неудобным действиям. Уменьшение объема информации может убрать перегрузку, но лишит пользователя важных деталей.

В предлагаемой модели reward включает оценку результата адаптации по таким признакам:

Критерий	Смысл для адаптации
Завершенность задачи	пользователь смог выполнить целевое действие
Снижение ошибок	после адаптации стало меньше неверных действий
Сокращение лишних шагов	взаимодействие стало менее затратным
Сохранение устойчивости UI	интерфейс не стал непредсказуемым

Критерий	Смысл для адаптации
Соответствие предпочтениям	конфигурация ближе к пользовательскому профилю

В контексте адаптации пользовательского интерфейса reward представляет собой сложный элемент модели, поскольку успешность изменения UI трудно оценить однозначно. Функция вознаграждения должна учитывать не только общие закономерности поведения пользователей, но и индивидуальные предпочтения конкретного пользователя.

4. Баланс *generality* и *individuality* в функции вознаграждения

Одним из ключевых элементов модели является баланс между *generality* и *individuality*. *Generality* отражает общие закономерности пользовательского поведения. *Individuality* отражает индивидуальные предпочтения конкретного пользователя. *Generality* важна, потому что интерфейс должен сохранять базовые свойства удобства. К ним относятся читаемость, понятная навигация, визуальная согласованность, доступность элементов и отсутствие перегрузки. Эти свойства не должны полностью зависеть от индивидуального профиля, потому что они формируют устойчивую основу пользовательского опыта. *Individuality* отражает индивидуальный уровень адаптации, связанный с устойчивыми особенностями взаимодействия конкретного пользователя с интерфейсом. В отличие от *generality*, данный компонент не описывает универсальные требования к удобству, а фиксирует персональные паттерны поведения, предпочтительные способы восприятия информации и реакции на выполненные изменения UI. Его значение состоит в том, что пользовательский опыт формируется не только общими принципами проектирования, но и накопленным опытом взаимодействия конкретного пользователя с системой. Если *individuality* не учитывается, адаптация остается усредненной и не позволяет точно настроить интерфейс под реальные сценарии работы пользователя.

При доминировании *generality* интерфейс остается устойчивым, но недостаточно персонализированным. При доминировании *individuality* система может стать слишком чувствительной к отдельным действиям пользователя и начать выполнять нестабильные адаптации. Оптимальная модель должна сочетать оба уровня. *Generality* задает безопасную и понятную основу интерфейса. *Individuality* уточняет интерфейс под конкретного пользователя.

Предлагаемая модель включает несколько связанных компонентов.

Компонент	Назначение
Модель состояния	описывает текущую конфигурацию UI и контекст пользователя
Модель действий	задает допустимые адаптации интерфейса
RL-агент	выбирает действие адаптации
Reward-модель	оценивает успешность действия
Модель <i>generality</i>	хранит общие UX-закономерности

Логика работы модели может быть описана следующим образом. Пользователь взаимодействует с веб-приложением. Система фиксирует состояние интерфейса и контекст. RL-агент выбирает действие адаптации. Интерфейс изменяется или остается прежним. После этого система получает сигналы результата: поведенческие данные, достижение цели, обратную связь и признаки соответствия предпочтениям. Reward-модель рассчитывает оценку.

В логике MDP процесс адаптации пользовательского интерфейса может быть представлен как последовательность: текущее состояние интерфейса и пользовательского контекста фиксируется системой; RL-агент выбирает допустимое действие адаптации; интерфейс переходит в новое состояние; результат изменения оценивается через *reward*; полученная оценка используется для уточнения дальнейшей политики выбора адаптаций. Веб-приложения имеют ряд особенностей, которые делают предложенную модель практически значимой. Используются разными группами пользователей,

поддерживают сложные сценарии и работают на разных устройствах. Один интерфейс должен оставаться понятным для новичка и достаточно эффективным для опытного пользователя. Для всех этих случаев важна не сама адаптация, а ее обоснованность. Изменение UI должно быть связано с задачей пользователя и оценено через reward. Если адаптация не дает положительного эффекта, она не должна закрепляться в политике агента. Веб-приложение также требует стабильности. Пользователь должен понимать, где находятся ключевые элементы и почему интерфейс изменился. Поэтому модель должна включать ограничения на частоту и масштаб адаптаций. Не каждое действие пользователя должно приводить к перестройке UI.

В отличие от подходов, где адаптация сводится к заранее заданным правилам или отдельным настройкам интерфейса, предлагаемая модель описывает адаптацию как обучаемый процесс. В ней пользовательская обратная связь рассматривается не как дополнительный комментарий, а как часть reward-механизма, влияющая на индивидуальный профиль и дальнейшую стратегию интерфейса. Такой подход позволяет соединить общие принципы UX с персональными реакциями пользователя. За счет этого интерфейс может сохранять устойчивость и одновременно адаптироваться к конкретному сценарию взаимодействия.

5. Заключение

Адаптивный интерфейс веб-приложения может быть представлен как интеллектуальная система, работающая в логике последовательного принятия решений. MDP позволяет формально описать состояние интерфейса, действия адаптации и reward. RL обеспечивает обучение агента на основе прошлых решений и их результатов. Ключевым элементом такой модели является функция вознаграждения. Она связывает техническое изменение интерфейса с пользовательским опытом. Баланс generality и individuality позволяет учитывать одновременно общие UX-закономерности и индивидуальные предпочтения пользователя. В отличие от подходов, где адаптация сводится к

заранее заданным правилам или отдельным настройкам интерфейса, предлагаемая модель описывает адаптацию как обучаемый процесс. В ней пользовательская обратная связь рассматривается не как дополнительный комментарий, а как часть reward-механизма, влияющая на индивидуальный профиль и дальнейшую стратегию интерфейса.

Предложенная модель показывает, что адаптация UI должна строиться не только на правилах или ручных настройках. Она может развиваться как замкнутый интеллектуальный контур, где каждое изменение интерфейса получает оценку и влияет на дальнейшую стратегию системы. Такой подход создает основу для веб-приложений, способных сохранять устойчивость интерфейса и одновременно подстраиваться под конкретного пользователя.

Литература

1. Курта П. А. Обзор способов построения динамических адаптивных интерфейсов и их интеллектуализация / П. А. Курта, К. Е. Израилов // Вестник Санкт-Петербургского университета ГПС МЧС России. – 2023. – № 4. – С. 119–132.
2. Winata G. I. Preference tuning with human feedback on language, speech, and vision tasks: A survey / G. I. Winata, H. Zhao, A. Das, W. Tang, D. D. Yao, S.-X. Zhang, S. Sahu // Journal of Artificial Intelligence Research. – 2025. – P. 2595–2661.
3. Langerak T. MARLUI: Multi-Agent Reinforcement Learning for Adaptive Point-and-Click UIs / T. Langerak, S. Christen, M. Albaba, C. Gebhardt, C. Holz, O. Hilliges // Proceedings of the ACM on Human-Computer Interaction. – 2024.
4. E. Mosqueira-Rey, E. Hernández-Pereira, D. Alonso-Ríos, J. Bobes-Bascarán, Á. Fernández-Leal. Mosqueira-Rey E. Human-in-the-loop machine learning: a state of the art // Artificial Intelligence Review. – 2023. – P. 3005–3054.
5. Moerl, T. M. Model-based Reinforcement Learning: A Survey / T. M. Moerland, J. Broekens, C. M. Jonker // Foundations and Trends in Machine Learning. – 2023. – № 1. – P. 1–118.