

«Человеко- компьютерное взаимодействие»»

Старший преподаватель кафедры «ПОИТ»
Малика Юлдашовна Доцанова

План:

- ▶ **Человеко-компьютерное взаимодействие**
- ▶ **Человеко-компьютерный интерфейс**
- ▶ **Различия в родственных областях**
- ▶ **Принципы разработки**
- ▶ **Методологии разработки**
- ▶ **Разработка дисплея**

Человеко-компьютерное взаимодействие

Человеко-компьютерное взаимодействие (HCI) — это изучение, планирование и разработка взаимодействия между людьми (пользователями) и компьютерами. Зачастую его рассматривают как совокупность науки о компьютерах, проектировании и других областей исследования.

Взаимодействие между пользователями и компьютерами происходит на уровне пользовательского интерфейса (или просто интерфейса), который включает в себя программное и аппаратное обеспечение; например, образы или объекты, отображаемые на экранах дисплеев, данные, полученные от пользователя посредством аппаратных устройств ввода (таких как клавиатуры и мыши) и другие взаимодействия пользователя с крупными автоматизированными системами.

Важным аспектом человеко-компьютерного взаимодействия является обеспечение удовлетворения пользователей. В связи с тем, что человеко-компьютерное взаимодействие изучается как с человеческой стороны, так и с компьютерной, то знания, полученные в ходе исследования, опираются как на человеческий фактор, так и на компьютерный.

С компьютерной стороны важны технологии компьютерной графики, операционных систем, языков программирования и среды разработки. С человеческой стороны, теория коммуникации, графический и промышленный дизайн, лингвистика, социология, когнитивная психология и такие человеческие факторы как удовлетворение пользователей.

Также имеет значение инженерия и проектирование. Благодаря междисциплинарному характеру человеко-компьютерного взаимодействия люди с разным уровнем подготовки вносят вклад в его успех. Иногда человеко-компьютерное взаимодействие называют как **человеко-машинное взаимодействие**, так и **компьютерно-человеческое взаимодействие**.

Основной задачей человеко-компьютерного взаимодействия является

улучшение взаимодействия между человеком и компьютером, делая компьютеры более удобными (usability) и восприимчивыми к потребностям пользователей. В частности, человеко-компьютерное взаимодействие занимается:

- ▶ методологией и развитием проектирования интерфейсов (т. е., исходя из требований и класса пользователей, проектирование наилучшего интерфейса в заданных рамках, оптимизация под требуемые свойства, такие как обучаемость и эффективность использования);
- ▶ методами реализации интерфейсов (например, программные инструментари, библиотеки и рациональные алгоритмы);
- ▶ методами для оценки и сравнения таких интерфейсов;
- ▶ разработкой новых интерфейсов и методов взаимодействия;
- ▶ развитием описательных и прогнозируемых моделей;
- ▶ теорией взаимодействия

Человеко-компьютерный интерфейс

Создание качественного человеко-компьютерного интерфейса, который можно назвать **точкой связи** между человеком и компьютером, есть конечная цель изучения человеко-компьютерного взаимодействия.

Обмен информацией между человеком и компьютером можно определить как **узел взаимодействия**. Узел взаимодействия включает в себя несколько аспектов:

Область задач:

- условия и цели, ориентированные на пользователя.

Область машины:

- среда, с которой взаимодействует компьютер, то есть ноутбук студента в комнате, в общежитии.

Области интерфейса:

- непересекающиеся области, касающиеся процессов человека и компьютера, не относящиеся к сфере взаимодействия.

Входящий поток:

- поток информации, который начинается в области задач, когда пользователь имеет несколько задач, которые требуют использования компьютера.

Выходной поток:

- поток информации, который возникает в машине.

Обратная связь:

- узлы взаимодействия, проходящие через интерфейс, оцениваются, модерируются и подтверждаются, так как они проходят от человека через интерфейс к компьютеру и обратно.

Различия в родственных областях

Человеко-компьютерное взаимодействие имеет различия с человеческим фактором в том, что человеко-компьютерное взаимодействие акцентирует своё внимание больше на пользователях, работающих предпочтительнее с компьютерами, чем с другими видами техники или артефактами.

Также человеко-компьютерное взаимодействие акцентирует своё внимание на реализации программного обеспечения и оборудования для поддержки человеко-компьютерного взаимодействия. Таким образом, *человеческий фактор* является более широким понятием; и человеко-компьютерное взаимодействие может быть охарактеризовано как человеческий фактор — хотя некоторые специалисты пытаются разделить данные области.

Также человеко-компьютерное взаимодействие отличается от *человеческого фактора* меньшим акцентированием внимания на задачах и процедурах и *гораздо меньшим* акцентированием на физическую нагрузку, истекающую из формы дизайна устройств интерфейса (таких как клавиатура и мышь).

Принципы разработки

При оценке текущего пользовательского интерфейса или разработке нового интерфейса следует иметь в виду следующие принципы разработки:

- С самого начала необходимо акцентировать своё внимание на пользователях и задачах: установить количество пользователей и определить подходящих пользователей; кто-либо никогда не использовавший интерфейс, либо тот, кто никогда не будет его использовать в будущем. Кроме того, необходимо определить, какие задачи и как часто будут выполнять пользователи.

- **Эмпирические измерения:** на ранней стадии провести тест интерфейса с реальными пользователями, которые используют интерфейс каждый день. Установить такие как: количество пользователей, выполняющих задачи, время выполнения задачи, и количество ошибок, сделанных в ходе выполнения задачи.

- Итеративное проектирование: после определения количества пользователей, поставленных задач, эмпирических измерений, выполните следующие шаги итеративной разработки:

- разработайте пользовательский интерфейс,
- проведите тестирование,
- проанализируйте результаты,
- повторите предыдущие шаги.

Повторяйте итеративную разработку до тех пор, пока не создадите практичный, удобный для пользователя интерфейс.

Методологии разработки

Разнообразные методики, излагающие техники проектирования человеко-компьютерного взаимодействия, начали появляться во времена развития данной области в 1980-х годах. Большинство методик разработки произошли от модели взаимодействия пользователей, разработчиков и технических систем. Ранние методики предлагали разработчикам при проектировании пользовательских интерфейсов рассматривать результаты исследований в таких областях как память и внимание.

Современные модели имеют тенденцию акцентировать внимание на постоянной обратной связи и диалоге между пользователями, разработчиками и инженерами, и прилагать усилия к тому, что технические системы крутятся вокруг желаний пользователей, нежели желания пользователей вокруг уже готовой системы.

Ориентированное на пользователя проектирование:

- разработка, ориентированная на пользователя, в данный момент является современной, широко практикующейся, суть которой заключается в том, что пользователи должны занимать центральное место в разработке любой компьютерной системы. Пользователи, разработчики и технические специалисты работают вместе, чтобы чётко выразить желания, потребности и границы, и создать систему, отвечающую этим требованиям. Ориентированные на пользователя проекты часто пользуются исследованиями этнографической среды, в которой пользователи будут работать с системой. Эта практика является аналогичной.

Принципы разработки пользовательского интерфейса:

- эти семь принципов могут рассматриваться в любое время, в любом порядке в течение всего времени разработки, это: привычность, простота, очевидность, допустимость, последовательность, структура и обратная связь.

Разработка дисплея

Дисплей предназначен для восприятия системных переменных и для облегчения дальнейшей обработки данной информации. Перед проектированием дисплея должны быть определены задачи, выполняемые данным дисплеем (например, навигация, управление, обучение, развлечение). Пользователь или оператор должен иметь возможность обработать любую информацию, которую генерирует и отображает система, поэтому информация должна отображаться в соответствии с принципами, которые обеспечивают восприятие и понимание.

13 принципов разработки дисплея

Кристофер Викенс выделил 13 принципов разработки дисплея в своей книге «An Introduction to Human Factors Engineering».

Эти принципы восприятия и обработки информации могут быть использованы для создания эффективного проекта дисплея. Сокращение количества ошибок, требуемого времени, повышение результативности и увеличение уровня удовлетворённости пользователей — это те из многих потенциальных выгод, которые могут быть достигнуты путём применения данных принципов. Некоторые принципы не могут быть применимы к некоторым дисплеям или ситуациям.

Некоторые принципы могут показаться противоречащими друг другу, и не существует доказательства, что один принцип является более важным, чем другой. Принципы могут быть адаптированы к конкретной разработке или ситуации. Функциональный баланс между принципами имеет важное значение для эффективной разработки.

Принципы, относящиеся к восприятию

1. Сделайте дисплей чётким.

- Читаемость дисплея является важным критерием в проектировании дисплея. Если символы или объекты отображаются нечётко, то пользователь не может эффективно их использовать.

2. Избегайте абсолютно строгих границ.

- Не просите пользователя определить уровень переменной на основе лишь одной сенсорной переменной (например, цвет, размер, громкость). Эти сенсорные переменные могут содержать множество различных уровней.

3. *Обработка сверху- вниз.*

- Сигналы воспринимаются и толкуются в соответствии с ожиданиями, сформированными на основе более раннего опыта пользователя. Если сигнал представлен вопреки ожиданиям пользователя, то потребуется больше его представления, чтобы доказать, что сигнал был понят верно.

4. *Чрезмерная выгода.*

- Если сигнал представлен более одного раза, больше шансов, что он будет понят верно. Это возможно сделать с помощью представления его в альтернативных физических формах (например, цвета, форма, голос и т. д.), так как избыточность не подразумевает повторения. Светофор является прекрасным примером избыточности, так цвет и положение являются избыточными.

4. *Чрезмерная выгода.*

- Если сигнал представлен более одного раза, больше шансов, что он будет понят верно. Это возможно сделать с помощью представления его в альтернативных физических формах (например, цвета, форма, голос и т. д.), так как избыточность не подразумевает повторения. Светофор является прекрасным примером избыточности, так цвет и положение являются избыточными.

5. *Сходства приводят к путанице.*

- Используйте отличающиеся элементы. Похожие сигналы будут приводить к путанице. Соотношение схожих признаков к различным признакам является причиной схожести сигналов. Например, A423B9 больше похоже на A423B8, чем 92 на 93. Ненужные похожие признаки должны быть удалены, а непохожие признаки должны быть выделены.

Принципы умозрительной модели

6. Принцип изобразительного реализма.

- Экран должен выглядеть как переменная, которую он представляет (например, высокая температура на термометре показана высшим вертикальным уровнем). Если есть множество составляющих, то они могут быть настроены так, как они будут выглядеть в среде, где они будут представлены.

7. Принцип движущейся части.

- Движущиеся элементы должны двигаться по той схеме и в том направлении, в каком это происходит в мысленном представлении пользователя, как оно движется в системе. Например, движущийся элемент на высотомере должен двигаться вверх с набиранием высоты.

Принципы, основанные на внимании

8.

*Минимизация
времени
доступа к
информации.*

- Когда внимание пользователя перемещается из одного места в другое в целях доступа к необходимой информации, то затрачивается много времени и усилий. Конструкция дисплея должна уменьшить данные затраты, так часто используемый источник должен находиться в ближайшей позиции. Однако не должна быть утрачена понятность.

9. Принцип
большого
количества
ресурсов.

- Пользователь может более просто обрабатывать информацию с разных ресурсов. Например, зрительная и слуховая информация может быть представлена одновременно, чем представлять всю зрительную и всю аудиоинформацию.

10. Принцип совместимости.

- Разделённое внимание между двумя источниками может быть необходимо для выполнения одной задачи. Эти источники должны быть мысленно взаимосвязаны и иметь мысленную близость. Время доступа к информации должно быть небольшим и это может быть достигнуто различными способами (например, близкое расположение, одинаковый цвет, узоры, формы и т. д.). Однако близость отображения может привести к путанице.

Принципы памяти

11. Замените память наглядной информацией: мировое знание.

- Пользователь не должен сохранять важную информацию исключительно в рабочей памяти или извлекать её из долговременной памяти. Меню/перечень могут помочь пользователю упростить использование памяти. Однако использование памяти иногда может помочь пользователю, так как избавляет от необходимости ссылаться на некоторые типы знаний в мире (например, компьютерный специалист скорее использовал бы прямые команды из памяти, чем обращался к руководству). Для эффективной разработки должны быть сбалансированы знания в голове пользователя и знания в мире.

12. Принцип предиктивной помощи.

- Проактивные действия, как правило, более эффективны, чем реактивные действия. Дисплей должен исключать ресурсоёмкие когнитивные задачи и заменить их более простыми задачами, чтобы сократить использование умственных ресурсов пользователя. Это позволит пользователю сконцентрироваться не только на текущей ситуации, но и также подумать о возможных ситуациях в будущем. Пример предиктивной помощи — дорожный знак, который информирует о расстоянии до пункта назначения.

13. Принцип совместимости.

- Старые особенности других дисплеев легко перенести в разработку новых дисплеев, если их разработки совместимы. Долговременная память пользователя будет срабатывать на выполнение уместных действий. В ходе разработки должен быть принят во внимание данный факт и учитывать совместимость между разными дисплеями.