

горелки. В результате прохождения газа через тангенциальное отверстие и камеру возникают акустические волны, положительно влияющие на процесс ионно-лучевого нанесения покрытий.

УДК 372

Бровка Д.С.

РАЗРАБОТКА ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

БНТУ, Минск

Научный руководитель Дробыш А.А.

Основой взаимодействия человека с компьютером является интерфейс, для проектирования которого используется два подхода либо с позиций человека-оператора, либо со стороны функциональных возможностей компьютера: инженерно-технический, когнитивный.

Инженерно-технический подход к созданию пользовательского интерфейса основан на предположении, что человек работает с компьютером подобно самому компьютеру, то есть, по определенному алгоритму.

Ввиду того, что инженерно-технический подход к проектированию интерфейса ориентирован на функциональные характеристики программы, пользователь, работающий с ней, вынужден «думать как разработчик».

Когнитивный подход, пришедший на смену алгоритмическому моделированию, рассматривает пользователя как центральную фигуру процесса взаимодействия с системой. Ориентация на характеристики пользователя, исследование перцептивных и когнитивных возможностей и ограничений человека позволили выявить закономерности взаимодействия человека с автоматизированной системой. Рассматривая процессы и закономерности восприятия, переработки информации и принятия решения, когнитивная психология выявила факторы, определяющие успешность выполнения задачи оператором. И это оказались не функциональные характеристики системы, как предполагалось инженерами раньше, а качество предоставления и управления информацией с точки зрения возможностей и ограничений человека.

Однако, как оказалось, анализа только процессов восприятия и переработки информации человеком недостаточно для

проектирования эргономичного интерфейса, поскольку он не позволяет определить состав и последовательность выводимой на экран информации. Это привело к появлению некоторого числа методологий дизайна UI, основанных на когнитивном подходе.

Существует множество рекомендаций от специалистов по проектированию пользовательского интерфейса. Эти рекомендации в большей или меньшей степени применимы как к созданию настольных и/или мобильных приложений, так и к веб-разработкам.

Ларри Константин, идеолог концепции дизайна, ориентированного на использование, в книге «SoftwareForUse», написанной им в 1999 г. совместно с Люси Локвуд, представил следующие принципы разработки интерактивных систем:

- Структурный принцип: проектирование интерфейса должно вестись целенаправленно, с использованием конструктивных решений, основанных на четких и последовательных моделях, узнаваемых для пользователя.

- Принцип простоты: дизайн должен быть простым, общие задачи должны быть понятны, общение между программой и человеком должно происходить на родном для него языке.

- Принцип видимости: все необходимые для решения конкретной задачи элементы интерфейса должны быть видимы и не должны отвлекать пользователя посторонней или избыточной информацией.

- Принцип обратной связи: дизайн должен информировать пользователей о выполняемых действиях, изменениях состояния или условий, об ошибках или исключениях.

- Принцип толерантности: дизайн должен быть гибким и терпимым к действиям пользователей, позволять отмену и повторное выполнение операций, а также предотвращать ошибки (где это возможно), интерпретируя все входные последовательности в разумные действия.

- Принцип повторного использования: интерфейс должен использовать согласованные внутренние и внешние компоненты, тем самым уменьшая для пользователей необходимость переосмысления или запоминания их (компонентов) назначения и поведения.

В 1994 году Якоб Нильсен, датский консультант по юзабилити, занимавшийся этим в фирмах IBM и SunMicrosystems, по результатам

факторного анализа 249 ранее выявленных проблем юзабилити представил набор эвристик, которые стоит учитывать при проектировании пользовательских интерфейсов:

1. Видимость состояния системы: система должна всегда и за приемлемое время должна реагировать на действия пользователя и информировать его о текущем состоянии работы.

2. Равенство между системой и реальным миром: система должна разговаривать с пользователем на его языке, используя слова, фразы и концепции, которые уже известны пользователю. Представление информации должно быть организовано в естественном и логичном порядке.

3. Свобода действий пользователя: пользователь должен иметь контроль над системой и возможность изменить текущее состояние программы путем отмены или повтора операций (Undo&Redo).

4. Последовательность и стандарты: принцип последовательности означает использование одних и тех же понятий и средств для отражения схожих образов и выполнения однотипных действий.

5. Предупреждение ошибок: система должна быть разработана так, чтобы минимизировать число ситуаций, в которых пользователь может совершить ошибку.

6. Понимание лучше, чем запоминание: все объекты, функции, действия должны быть видны пользователю.

7. Гибкость и эффективность использования: чтобы интерфейс программы был одинаково удобен как для новичков, так и для опытных пользователей, необходимо обеспечить альтернативные способы работы с ним.

8. Эстетичный и минималистичный дизайн: диалоги не должны содержать нерелевантную или редко используемую информацию.

9. Распознавание и исправление ошибок: «Помогайте пользователю распознавать, диагностировать и исправлять ошибки» – говорит Якоб Нильсен и поясняет, что сообщения об ошибках должны быть выражены простым языком (без кодов), точно описывать проблему и предлагать конструктивное решение для нее.

10. Справка и документация: лучшая система та, которая может быть использована без какой-либо документации.

Бен Шнейдерман, американский исследователь в области человеко-машинного взаимодействия, в своей книге «Designing the User Interface» сформулировал восемь «золотых» правил, которые кратко можно представить в следующем виде:

1. Будьте последовательны: используйте одинаковые действия, названия, элементы управления в идентичных или похожих ситуациях.

2. Учитывайте возможности опытных пользователей: представьте им альтернативные способы управления программой с помощью «горячих» клавиши, макросов и т.п.

3. Используйте обратную связь: программа должна реагировать на каждое действие оператора.

4. Создавайте законченные диалоги: сформируйте последовательные действия оператора в логические группы с началом, серединой и концом.

5. Используйте простые процедуры обработки ошибок: насколько возможно, спроектируйте систему так, чтобы пользователь не мог допустить серьезных ошибок, а при обнаружении ошибки предложите простые и понятные механизмы ее обработки.

6. Обеспечьте простой механизм отмены действий: такая возможность уменьшает беспокойство пользователей, так как они знают, что ошибочные действия могут быть отменены. Единицей обратимости может быть разовая акция, ввод данных или целая группа действий.

7. Создайте впечатление, что пользователь управляет всеми процессами: спроектируйте систему так, чтобы оператор был инициатором действий, а не ведомым.

8. Уменьшите загрузку кратковременной памяти: особенности человеческой памяти накладывают ограничения на количество, размеры и скорость чередования элементов управления.

В целом разработка пользовательского интерфейса есть сложный, многогранный процесс, требующий сложных специальных знаний, умений и навыков.