

УДК.004.031.42 + 332.1

WEB-ИНТЕРФЕЙС УПРАВЛЕНИЯ СИСТЕМАМИ ЖИЗНЕОБЕСПЕЧЕНИЯ

Губтор А.О., Дмитриев В.Е., Лысюк А.А., Соколов А.М.

*Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования «Новосибирский государственный
педагогический университет», г. Новосибирск*

Аннотация

В статье описываются интерфейсы управления вычислительными системами, преимущества web-интерфейсов.

Ключевые слова: web-интерфейс, http, клиент-сервер приложения.

Интерфейсы управления. Если стоит задача разработки приложения для удаленного доступа (управление или мониторинг), то эта задача в конечном итоге сводится к задаче эффективной передачи данных от одного устройства к другому. Попутно мы должны решить ряд проблем. Одна из этих проблем это способ взаимодействия пользователя с приложением (интерфейс). Если мы говорим о интерфейсах (способы взаимодействия пользователя с приложением), то выделяют:

- интерфейс для программирования приложения (API),
- интерфейс командной строки (CLI),
- графический интерфейс пользователя (GUI).

Здесь мы пока не затрагиваем способ обмена данными между устройствами. Предположим, что приложение действует на локальном компьютере, и производит какие либо вычисления. С точки зрения разработки такого приложения, наиболее простым является командный интерфейс, так как программа может принимать передаваемые ей параметры через устройство ввода, обрабатывать их, и выводить результат в устройство вывода. Для программиста и профессионала в данной области это наиболее удобный и простой способ взаимодействия с программой. Но если мы говорим о среднем статистическом пользователе который не хочет или у которого нет времени разбираться с системой команд приложения и ключами, то конечно такой способ управления не подходит, для такого пользователя гораздо удобней и понятней пользоваться графическим интерфейсом на котором в виде кнопок и указателей обозначены основные команды и функции приложения и путем нажатия на кнопки, а так же путем ввода данных в специальные формы можно получить желаемый результат.

Необходимо заметить, что за подобные удобства необходимо заплатить дополнительной нагрузкой на аппаратные средства компьютера. Графический интерфейс, поддержка масштабируемых шрифтов, поддержка многозадачности и т.д. требуют большой мощности процессора, значительной оперативной памяти и дискового пространства.

Графический пользовательский интерфейс (ГПИ, англ.- Graphical User Interface- GUI) появился как ответ на не отличающийся удобством и понятностью интерфейс командной строки. Командная строка требует ввода команд с клавиатуры. Так как команды необходимые для ввода могут быть многочисленны, со сложными операторами и ключами, и составлять целые предложения слов и символов, это позволяет достигнуть большой эффективности и гибкости, при условии, что все эти команды и способы их применения хорошо изучены, но достижение такого уровня подготовлен-

ности обычно занимает продолжительное время потому что командные слова не так легко найти, и фактически каждый производитель программы использовал (и использует) для команд и ключевых слов собственные слова и фразы, иногда это могут быть акронимами английских слов, иногда это просто сокращённые слова. К тому же использование командной строки может в некоторых случаях замедлить работу или вызывать частые ошибки, например при вводе очень длинных команд включающих много параметров и/или несколько различных имен файлов за один раз. Графический интерфейс предоставляет пользователю интуитивно понятные виджеты (кнопки, формы, чек-боксы) при выборе которых (с помощью указывающих устройств, мыши или клавиатуры), наступает событие и программа выполняет необходимые действия. Большинство современных приложений и операционных систем предоставляют, как графический так и командный пользовательский интерфейс [5].

Понятный графический пользовательский интерфейс предоставляет совершенно не подготовленному пользователю (единственное требование это умение читать) набор команд и опций, которые всегда видны на экране. Так же основные элементы графического интерфейса всегда одни и те же, и пользователю приходится меньше учиться, чтобы освоить данное приложение, и сосредоточится на решении тех проблем, для решения которых он использует компьютер.

Клиент – сервер. Клиент-сервер (англ. Client-server) — вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг (сервисов), называемыми серверами, и заказчиками услуг, называемыми клиентами. Нередко клиенты и серверы взаимодействуют через компьютерную сеть и могут быть как различными физическими устройствами, так и программным обеспечением. Для реализации нашего проекта а именно панели управления была выбрана немного расширенная схема клиент-сервер [6]. С одной стороны на объекте, модули-клиенты собирают необходимые данные и передают их своему серверу, на сервере приложение обрабатывает данные оформляя их в XML пакеты и отправляя по протоколу HTTP на второй веб сервер (презентация), который предоставляет данные в удобном и понятном виде пользователю, а так же дает возможность через графические элементы управления формировать и отправлять на шлюз XML пакеты уже содержащие управляющие команды для конечных устройств (модули управления) [1,2,3].

Преимущества схемы клиент сервер:

- Отсутствие дублирования кода программы-сервера программами-клиентами.
- Так как все вычисления выполняются на сервере, то требования к компьютерам на которых установлен клиент снижаются.
- Все данные хранятся на сервере, который, как правило, защищён гораздо лучше большинства клиентов. На сервере проще обеспечить контроль полномочий, чтобы разрешать доступ к данным только клиентам с соответствующими правами доступа.
- Позволяет объединить различные клиенты. Использовать ресурсы одного сервера часто могут клиенты с разными аппаратными платформами, операционными системами и т. п.
- Позволяет разгрузить сети за счёт того, что между сервером и клиентом передаются небольшие порции данных.

Так же у данной схемы есть и *недостатки*.

- неработоспособность сервера может сделать неработоспособной всю вычислительную сеть. (Неработоспособным сервером следует считать сервер, производительности которого не хватает на обслуживание всех клиентов, а также сервер, находящийся на ремонте, профилактике и т. п.)

- Поддержка работы данной системы требует отдельного специалиста — системного администратора.[6]

- Относительно высокая стоимость оборудования.

Выбор HTTP. Определившись с интерфейсом необходимо выбрать транспортное средство для передачи данных со шлюза пользователю и базе данных. Для решения этой задачи подходит протокол HTTP по нескольким причинам.

1. Протокол HTTP является стандартным протоколом для обмена данными [4].

2. Протокол HTTP поддерживает передачу мультимедиа информации (текст, видео, аудио и т.д.)

В связи с выбором данного протокола решается еще несколько проблем. Часто на компьютере пользователя установлено множество приложений, и если разрабатывается приложение для компьютера конечного пользователя то нагрузка на его компьютер только возрастает т.к у подобных приложений с нашей точки зрения имеется ряд недостатков.

- Высокие требования к компьютеру пользователя

- Процессорное время

- Оперативная память

- Постоянная память

- Проблема доступа к данным и к панели управления (Если пользователь находится не дома и у него нет под рукой устройства с установленным приложением, то он не будет иметь доступ к панели управления и к данным).

В случае перенесения всей бизнес логики на удаленный сервер пользователю предоставляется только интерфейс этого приложения в виде веб страницы. Это значит, что веб-браузер может быть единственным приложением установленным на компьютере пользователя. Необходимое условие доступа это наличие интернет подключения (заметим что без него ни о каком удаленном управлении конечно речи не идет). Так как интернет получает все большее распространение, то можно предсказать, что интернет в ближайшее время будет так же распространен как и телевидение, а учитывая возможности радио сетей (Wi-Fi, GSM и т.д) то возможно и более широко. В то же время веб-браузер теперь является стандартным приложением, идущим в комплекте с большинством операционных систем.

Литература

1. Соколов, А.М., Попов Е.А., Дмитриев В.Е., Губтор А.О., Лысюк А.А. Свидетельство о государственной регистрации программы для ЭВМ №2013611674. Программа автоматизированного управления и мониторинга систем жизнеобеспечения и энергосбережения, заявка № 2012660709 от 06 декабря 2012г. Зарегистрировано в Реестре программ для ЭВМ 30 января 2013г.
2. Соколов, А.М. Системы сбора и мониторинга информации. Молодежь XXI века: образование, наука, инновации: материалы 1 науч.-пркт. конф. студ. ФТиП. - Новосибирск: СИЦ "Гаудеамус" ФГБОУ ВПО "НГПУ", 2012.
3. Таненбаум, Э. Компьютерные сети. 4-е изд. / — СПб.: Питер, 2003. — 992 с: ил. — (Серия «Классика computer science»).
4. Lawrence, Eric (22 October 2005). "[Upcoming HTTPS Improvements in Internet Explorer 7 Beta 2](#)". [Microsoft](#). Retrieved 12 May 2009.
5. Martinez, W. L. (2011), Graphical user interfaces. WIREs Comp Stat, 3: 119–133. doi: 10.1002/wics.150.
6. URL: <http://ru.wikipedia.org/wiki/Клиент-сервер> (дата обращения 4 мар. 2013 г.).URL: <http://ru.wikipedia.org/wiki/Клиент-сервер> (дата обращения 4 мар. 2013 г.)