

Лекция 15.

Кинетические формы взаимодействия с пользовательскими интерфейсами

Цель лекции: в данной лекции рассматриваются кинетические формы взаимодействия с пользовательскими интерфейсами. Описывается интерфейс взаимодействия приложений WEB-браузера и WEB-сервера. Рассматриваются составляющие программно-аппаратных интерфейсов на основе протоколов уровня приложений.

Вопросы для рассмотрения: Интерфейсы взаимодействия уровня приложений.

Выполнение интерфейса взаимодействия web - приложений.

Основные термины: WEB-браузер, WEB-сервер, HTTP, URL, URI, HTML, интерфейс

Введение

Рассмотрим интерфейс взаимодействия двух приложений WEB-браузера и WEB-сервера (поскольку эти приложения обычно располагаются на разных машинах и, соответственно, на разных программно-аппаратных платформах используем термин программно-аппаратный интерфейс). При реализации интерфейса взаимодействия WEB - приложений используется протокол HTTP (Hypertext Transfer Protocol - протокол передачи гипертекста), который представляет собой протокол прикладного уровня и обеспечивает возможность доступа к разнообразной информации, размещенной в сети WWW - World Wide Web.

Протокол HTTP обладает высокопроизводительными механизмами тиражирования информации, независимо от типа представления данных. Протокол построен по объектно-ориентированной технологии и может использоваться для решения различных задач, например, управления распределенными информационными системами.

Способность хранить и представлять данные разнообразных форматов (изображения, видео, аудио) делает сеть WWW с используемым HTTP уникальным средством размещения информации. В настоящее время протокол HTTP используется системой WWW качестве одного из основных протоколов. С учетом этого рассмотрим подробнее методы работы протокола HTTP. Протокол HTTP позволяет получать доступ к информационным ресурсам и сервисам WWW-серверов. Для унификации доступа к многофункциональным ресурсам сети WWW-серверы поддерживают комплекс интерфейсов, позволяющих структурировать уровни

и методы доступа к сетевым ресурсам. По сути, каждый из интерфейсов представляет собой объект сети со своими

размещенной в сети WWW, применяются специальные приложения, называемые Web-браузерами. Согласованное взаимодействие объектов (клиентских и серверных) и составляет понятие программного интерфейса.

Рассмотрим составляющие программно-аппаратных интерфейсов на основе протоколов уровня приложений.

URI (Uniform Resource Identifier, Идентификатор ресурса), **URL** (Uniform Resource Locator, Местонахождение ресурса), **URN** (Uniform Resource Name, Имя ресурса) - разные аспекты идентификации одного и того же сервиса, определяющие тип, метод доступа и расположение узла сети, на котором находятся ресурс, доступный через сеть Интернет.

Этот сервис состоит из трех частей:

1) **Схема**. Идентифицирует тип сервиса, через который можно получить доступ к сервису, например, WWW-сервер.

2) **Адрес**. Идентифицирует адрес (хост) ресурса, например, *www.ripn.net*.

3) **Имя или путь доступа**. Идентифицирует полный путь к ресурсу на выбранном хосте, который мы хотим использовать для доступа к ресурсу, например, */home/images/image 1. gif*. Например, файл *readme. txt*, расположенный на сайте Microsoft (WWW-сервере), представляет собой ресурс с идентификатором: *http://www.microsoft.com/readme. txt*. Это означает, что для обращения к ресурсу должен использоваться протокол HTTP, (схема доступа отделена двоеточием " : " и указывает название использованного протокола), следующие два слэша отделяют адрес сервера *www.microsoft.com*; а также) имя файла */readme. txt*.

Как правило, когда имеют в виду компьютер, на котором расположен ресурс, используют значение URL или URN, а когда обозначают ресурс полностью (тип, хост, путь) используют URI. Нет ошибки, если используется одно обозначение вместо другого, но обязательно следует пояснить, что оно значит в контексте.

Идентификатор URI может содержать не только имя ресурса, но и параметры, необходимые для его представления. Имя ресурса отделяется от

строки параметров символом "?". Строка параметров состоит из символьных групп с постоянной структурой (лексем), разделяемых символом "&", каждая такая лексема состоит из имени параметра и его значения, разделенных символом "=", символ пробела " " заменяется знаком "+". Символы лексем, не входящие в набор символов ASCII, заменяются знаком "%" и шестнадцатеричным значением этого символа. Для указанного ресурса вся строка параметров является одним строковым параметром, поэтому тип, порядок следования или уникальность имен отдельных параметров строки не существенны. Например:

`http://www.exe.com/bm/scrshell.run?`

`in=10&go=ok+and+ok&event=1&event=2` Этот идентификатор URI содержит 4 параметра, три из которых - числовые, а два имеют одно имя. Анализ и разбор значений отдельных параметров целиком возлагается на идентификатор URI, в данном примере на ресурс `scrshell.run`.

HyperText Markup Language (HTML) - это язык описания информации, хранимой в сети WWW.HTML-файл может содержать специальные коды, обозначающие присоединенную графическую, видео или аудио информацию или исполняемые коды среды отображения информации (Web-браузер - Java Script, Java). Для языков Java и JavaScript приложение Web-браузер представляет операционную систему или среду, в которой они выполняются, а Web-страница является ресурсом, выделенным для их работы. Эти языки не строят Web- страницу по данным пользователя, а используют ее как платформу для своих действий и действий пользователя. Когда Web-браузер получает доступ к этому файлу, он сначала интерпретирует закодированную в HTML-файле информацию, а затем в соответствующей форме представляет эту информацию пользователю.

Буквы "HT" в названии протокола HTML обозначают "HyperText" - основную концепцию размещения информации в сети WWW. Документы HyperText содержат специальные связи, которые называются гиперссылками (hyperlinks) и размещаются в тексте документа. Гиперссылки позволяют

пользователю не только переходить от одной части этого документа к другой, но и обращаться к другим связанным документам, размещенным в сети WWW.

Common Gateway Interface (CGI) - это стандарт расширения функциональности WWW, позволяющий WWW-серверам выполнять программы, аргументы которых может определять пользователь. Интерфейс CGI расширяет возможности пользователя и позволяет ему выполнять программы, ассоциированные с данной Web-страницей, предоставляя таким образом возможность получения динамической информации из WWW-сервера. Например, пользователь такого WWW-сервера может получить самую последнюю информацию о погоде, выполнив программу, которая запрашивает прогноз погоды на текущий момент из базы данных. Интерфейс CGI в основном играет роль шлюза между WWW-сервером и внешними исполняемыми программами. Он получает запрос от пользователя, передает его внешней программе и затем возвращает результаты пользователю через построенную динамически Web-страницу. При этом построенные Web-страницы могут коренным образом отличаться друг от друга, поскольку они формируются в прямой зависимости от параметров, определяемых пользователем.

Механизм интерфейса CGI также является универсальным и может передавать данные между любыми WWW-серверами. Поскольку интерфейс CGI основан на исполняемых файлах, нет ограничений на тип программы, которая будет в нем исполняться. Программа может быть написана на любом из языков программирования, позволяющих создавать исполняемые модули. CGI- программа также может быть написана с использованием командных языков операционных систем, таких как Perl или Shell.

В настоящее время широко используется технология активных серверных страниц ASP (Active Server Pages). По сути, эта технология представляет применение того же самого стандарта CGI, только на уровне объектно- ориентированного подхода к построению Web-страниц.

Контрольные вопросы

1. Что понимается под кинетическими формами взаимодействия с пользовательскими интерфейсами?
2. Опишите интерфейс взаимодействия приложений WEB-браузера и WEB-сервера.
3. Какие составляющие программно-аппаратных интерфейсов на основе протоколов уровня приложений существуют? Опишите каждый из них.

Литература:

1. Купер А. т.б. Об интерфейсе: основы проектирования взаимодействия. 4-е изд. – СПб: Питер, 2018. – 720 б.
2. Норман Д. Дизайн привычных вещей. – М: Манн, Иванов и Фербер, 2019. – 384 б.
3. The Encyclopedia of Human-Computer Interaction, 2nd Ed. [электрондық басылым] – URL: <https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed> (соңғы алышған 8/1/2020)
4. Sharp Н. Interaction Design: Beyond Human-Computer Interaction. 5th Ed. – Wiley, 2019. – 656 p.
5. Shneiderman В. et.al. Designing the user interface: strategies for effective human-computer interaction. 6th Ed. – Pearson, 2016. – 616 p.
6. Тидвелл Д. т.б. Разработка пользовательских интерфейсов. 2-е изд. – М: Питер, 2011 – 480 б.
7. Круг С. т.б. Веб-дизайн или «не заставляйте меня думать». – СПб.: Символ Плюс, 2008. – 224 б.
8. Нильсен Я. т.б. Веб дизайн. СПб: Символ Плюс, 2006. – 512 б.
9. Уильямс Р. т.б. Не дизайнерская книга о дизайне. – СПб: Весь, 2004. – 128 б.
10. <http://appscamp.io/> – Онлайн-курс который дает начальное понимание разработки на HTML и мобильных платформах.
11. <http://phonegap.com/book/> – Список книг по разработке HTML и мобильных приложений с помощью фреймворка PhoneGap.
12. <http://creator.ionic.io/> – HTML фреймворк пользовательского интерфейса для мобильных приложений.