

УДК 004.273

АРХИТЕКТУРА WEB-ОРИЕНТИРОВАННЫХ САПР

Г. Д. Дмитриевич,

доктор техн. наук, профессор

А. А. Мохсен,

аспирант

А. И. Ларистов,

канд. техн. наук, доцент

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»

Показана архитектура современных Web-приложений, которые представляют собой коллекцию элементов Web-узла, программно выполняющих какие-либо действия и являющихся основой Web-ориентированных САПР. Web-приложения (Web-САПР) создаются таким образом, чтобы они выполнялись на Web-серверах и использовались в качестве пользовательского интерфейса Web-браузеры. Обычно Web-приложения создаются как приложения в архитектуре «клиент–сервер», но серверная часть может иметь различные архитектурные решения. Приведем пример архитектуры конкретной Web-ориентированной САПР.

Ключевые слова — Web-приложение, Web-ориентированная САПР, архитектура Web-приложения, архитектура Web-ориентированной САПР, архитектура клиент–сервер, образовательные порталы.

Одним из направлений развития систем автоматизированного проектирования в настоящее время является разработка распределенных архитектур и построение на их основе Web-ориентированных САПР. Реализация подобной распределенной архитектуры САПР требует создания специального Web-приложения, обеспечивающего запуск и синхронизацию подсистем на стороне клиента и на стороне сервера, а также пересылку данных между клиентскими и серверными подсистемами. Рассмотрим основные методы построения Web-приложений.

Изначально World Wide Web (WWW) создавалась как «пространство для обмена информацией, в котором люди и компьютеры могут общаться между собой» [1]. Поэтому первые Web-приложения представляли собой примитивные файловые серверы, которые возвращали статические HTML-страницы запросившим их клиентам. Таким образом, Web начиналась как документо-ориентированная сеть.

Следующим этапом развития Web стало появление понятия приложений, которые базировались на таких интерфейсах, как CGI (или FastCGI), а в дальнейшем — на ISAPI. Common Gateway Interface (CGI) — это стандартный интерфейс с серверами, позволяющий выполнять серверные приложения, вызываемые через URL. Входной информацией для таких приложений служило

содержимое HTTP-заголовка (и тело запроса при использовании протокола POST). CGI-приложения генерировали HTML-код, который возвращался браузеру. Основной проблемой CGI-приложений было то, что при каждом клиентском запросе сервер выполнял CGI-программу в реальном времени, загружая ее в отдельное адресное пространство.

Появление Internet Server API (ISAPI) позволило не только решить проблемы производительности, которые возникали с CGI-приложениями, но и предоставить в распоряжение разработчиков более богатый программный интерфейс. ISAPI DLL можно было ассоциировать с расширениями имен файлов через специальную мета-базу. Эти два механизма (CGI и ISAPI) послужили основой создания первого типа Web-приложений, в которых в зависимости от каких-либо клиентских действий выполнялся серверный код. Таким образом, стала возможной динамическая генерация содержимого Web-страниц и наполнение Web перестало быть чисто статическим.

Интерфейс ISAPI — это особенность Microsoft Internet Information Server. ISAPI-приложения представляют собой динамические загружаемые библиотеки (DLL), которые выполняются в адресном пространстве Web-сервера. У других Web-серверов через некоторое время также появилась возможность выполнять приложения, реализо-

ванные в виде библиотек [2]. В случае Web-серверов Netscape этот программный интерфейс назывался NSAPI (Netscape Server API). У довольно популярного Web-сервера Apache также имеется возможность выполнять Web-приложения, реализованные в виде библиотек; такие библиотеки называются Apache DSO (Dynamic Shared Objects).

Отметим, что с ростом объема используемых данных и числа посетителей Web-сайтов возрастают и требования к надежности, производительности и масштабируемости Web-приложений. Следующим этапом эволюции подобных приложений стало отделение бизнес-логики, реализованной в Web-приложении, а нередко и сервисов обработки данных и реализации транзакций от его интерфейса. В этом случае в самом Web-приложении обычно остается так называемая презентационная часть, а бизнес-логика, обработка данных и реализация транзакций переносятся в сервер приложений в виде бизнес-объектов.

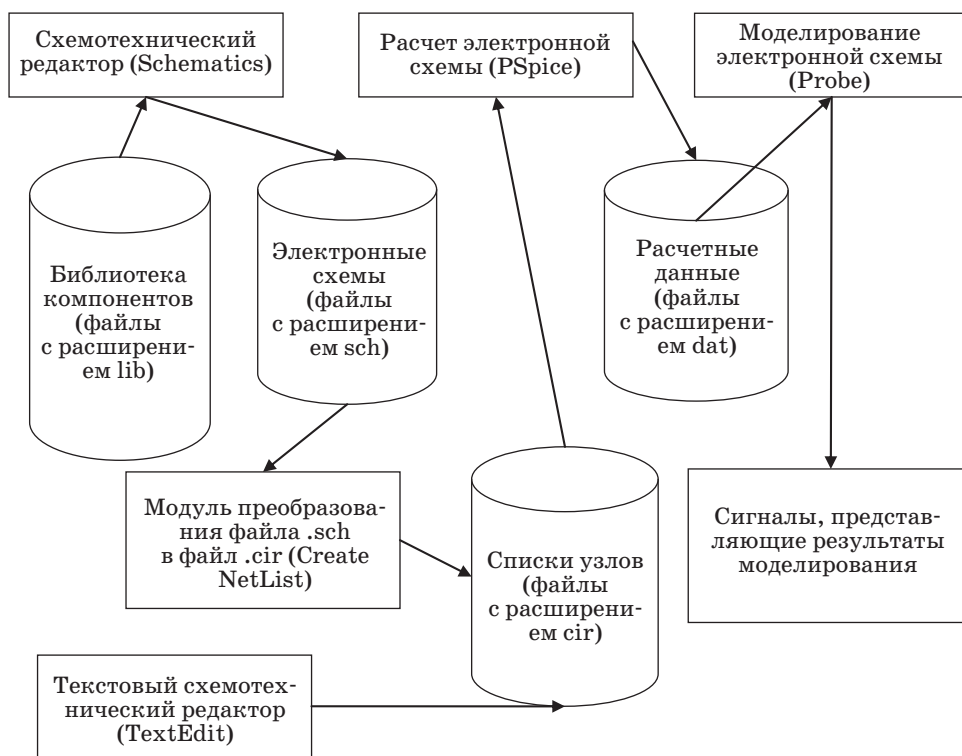
Следующим шагом эволюции Web-приложений, помимо доступа к корпоративным данным и данным партнеров, стало получение доступа к корпоративным приложениям. Для интеграции Web-приложений с внутренними информационными системами предприятий и приложениями, обеспечивающими взаимодействие с клиентами и партнерами, используются специальные решения, называемые корпоративными пор-

талами. Нередко в содержимое портала включают средства управления информационным наполнением Web-сайта, поскольку объем данных, доступных пользователям с помощью сайтов крупных компаний и порталов, сейчас таков, что управление этими данными «вручную» не представляется возможным [3].

Решение многих описанных выше задач, возникающих при создании современных Web-приложений, теперь начинает возлагаться на Web-сервисы — не зависящие от платформы, объектной модели и клиента программные компоненты, которые можно вызывать из клиентских Web-приложений (а также из самих Web-сервисов) через основанный на протоколе HTTP и языке XML протокол SOAP. Для описания Web-сервисов используется XML-подобный язык WSDL, а для организации реестров Web-сервисов, в которых разработчики и компании могут искать необходимые им сервисы, а также публиковать данные о своих сервисах, — интерфейс UDDI.

Обзор возможных архитектур построения Web-приложений позволяет сделать вывод о реализации распределенных САПР на основе технологии ASP.NET, предложенной фирмой Microsoft в среде Visual Studio. Рассмотрим применение данной технологии при разработке Web-ориентированной САПР электронных схем.

Традиционная архитектура САПР (рис. 1) включает отдельные подсистемы, написанные на



■ Рис. 1. Традиционная архитектура САПР

процедурных или объектно-ориентированных языках программирования, имеющих мощные средства для работы с различными структурами данных, таких как списки, очереди, стеки, деревья, графы и т. д. Обмен данными между подсистемами обеспечивается через общую область памяти, через систему двоичных файлов, посредством специализированных языков описания (текстовые файлы).

На настоящий момент актуальными являются разработка и адаптация специальных Internet-ориентированных версий настольных САПР, позволяющих обеспечить автоматизацию процесса проектирования объектов различной физической природы в масштабах предприятия.

Рассмотрим архитектуру Internet-версии подобной САПР на примере САПР электронных схем Design Lab корпорации MicroSim, получившую наибольшее распространение среди разработчиков радиоэлектронной аппаратуры.

Основу системы Design Lab составляет программа PSpice, которая является наиболее известной модификацией программы схемотехнического моделирования SPICE (Simulation Program with Integrated Circuit Emphasis), разрабо-

танной в начале 70-х гг. в Калифорнийском университете г. Беркли. Она оказалась очень удачной, интенсивно развивается и де-факто стала эталонной программой моделирования аналоговых устройств.

В состав САПР Design Lab входят следующие подсистемы:

Schematics — интегрированная управляющая подсистема и графический редактор электронных схем;

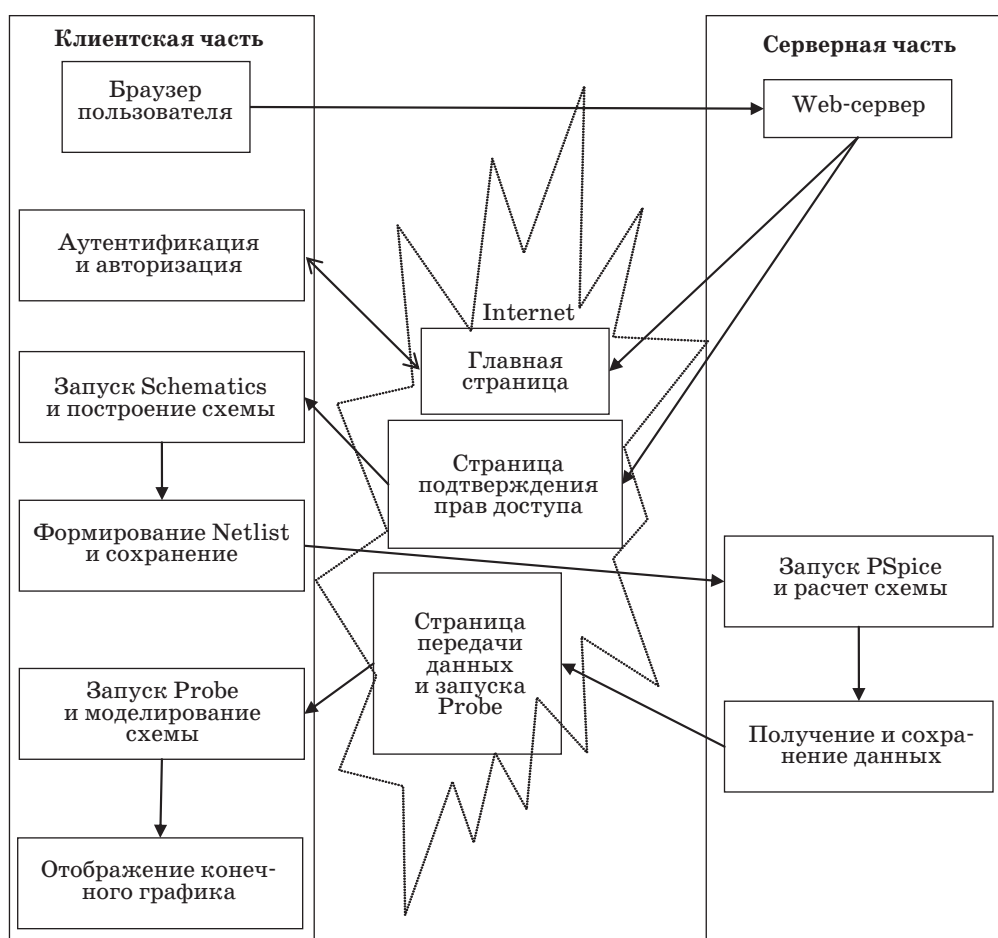
PSpice — проектирующая подсистема, выполняющая построение математической модели и численный анализ схемы;

Probe — графический постпроцессор, ориентированный на отображение в виде графиков результатов моделирования;

StmEd — редактор входных сигналов, позволяющий задавать произвольную форму возмущающих воздействий [4];

Parts — подсистема расчета параметров моделей схемных компонентов.

Для передачи данных между подсистемами Schematics и PSpice используется текстовый файл с описанием схемы на входном языке системы.



■ Рис. 2. Архитектура Web-ориентированной САПР электронных схем

Передача данных моделирования из PSpice в Probe осуществляется с помощью двоичного файла.

В процессе работы САПР активно используются библиотеки параметров схемных компонентов, представляющие собой текстовые файлы на языке задания описания моделей и параметров компонентов схем.

Для адаптации работы системы Design Lab в сети Internet необходима разработка новой распределенной архитектуры САПР с разделением функций между клиентом и сервером, чтобы добиться оптимальной производительности в условиях низкоскоростных каналов Internet и лимитированных ресурсов Web-серверов [5]. Так, предварительную обработку и ввод данных, отправляемых серверу, имеет смысл выполнять на стороне клиента. Это позволит исключить, например, повторные передачи неправильно составленных схем. Графическое представление результатов запроса также стоит выполнять на стороне клиента, что существенно сократит объем данных, передаваемых по сети. Моделирование схем и доступ к библиотекам параметров схемных компонентов целесообразно обеспечить за счет ресурсов сервера. Отметим, что сеть Intranet — отличная платформа для работы с информацией внутри предприятия. Современный Web-браузер доступен для любой клиентской системы.

Архитектура Web-ориентированной САПР электронных схем представлена на рис. 2.

В настоящее время стали появляться Web-ориентированные САПР. Они представляют со-

бой приложения, разделенные на 3 модуля — клиентский, обслуживающий и проектирующий. Клиентский модуль, выполняемый на компьютере пользователя, осуществляет формирование данных для их передачи на сервер. Обслуживающий модуль через Web-страницы запускает части клиентского модуля. Проектирующий модуль, выполняемый на сервере, производит обработку поступивших данных (см. рис. 2). Intranet ориентирован, как правило, на применение в рамках одного компактного или распределенного предприятия и отличается высокой безопасностью и скоростью работы. Используется для решения задач по автоматизации документооборота, информационному сопровождению бизнес-процессов, поиска и совместного доступа к данным и документам организации и имеет шлюзы для подключения в Internet.

В данной статье мы изучили эволюцию архитектуры Web-решений, начиная от простейших хранилищ HTML-страниц и заканчивая современными корпоративными решениями, интегрированными с корпоративными информационными системами и информационными системами партнеров. Обсудили задачи, возникающие на каждом этапе развития Web-приложений и технологий, их решающие, включая CGI, ISAPI; взаимодействие с серверами приложений и с базами данных; создание и применение Web-сервисов, основанных на XML. Предложили архитектуру Web-ориентированной САПР на основе популярной САПР электронных схем Design Lab корпорации MicroSim.

Литература

1. Berners-Lee T. WWW: Past, present, and future // Computer. Oct. 1996. Vol. 29. N 10. P. 69–77.
2. <http://supportline.microfocus.com/Documentation/books/nx30books/piisapcn.htm> (дата обращения: 10.11.2009).
3. <http://zone.ni.com/devzone/cda/epd/p/id/6363> (дата обращения: 02.12.2009).
4. Разевиг В. Д. Система схемотехнического моделирования и проектирования печатных плат Design Center (PSpice). — М.: СК Пресс, 1996. — 272 с.
5. Разевиг В. Д. Система схемотехнического моделирования Micro-CAP V. — М.: СОЛОН, 1997. — 273 с.