

## МОДЕЛИ ОБРАБОТКИ ДАННЫХ И ПОДГОТОВКА СПЕЦИАЛИСТОВ В ОБЛАСТИ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

А.С. Деревянко, к.т.н. М.Н. Солошук  
(представил д.т.н., проф. Л.М. Любчик)

В статье обосновывается необходимость ведения подготовки специалистов в области информационных технологий с учетом того, в какой модели обработки данных будут работать эти специалисты. Рассматриваются три модели: персональная, офисная и промышленная. Для каждой модели определяются основные характеристики: состав задач, спектр применяемых программных и аппаратных средств, технология изготовления программных продуктов, профессиональные качества и ментальность специалистов.

Начавшаяся более 15 лет назад "персональная революция" в вычислительной технике коренным образом изменила представление об автоматизированной обработке данных (АОД) у людей, не являющихся специалистами в информационных технологиях (ИТ). В "обывательском" представлении инструмент АОД - это прежде всего персональный компьютер (ПК) на рабочем столе, который хранит все необходимые для работы данные и быстро выдает ответ на любой вопрос или обеспечивает выход в сеть Internet, которая опять-таки быстро выдает ответ на любой вопрос. Однако, в действительности сфера применений вычислительной техники не ограничивается только ПК и персональными технологиями. Мы разделяем мнение ряда авторов (например, [1]), согласно которым можно определить по крайней мере три "модели" АОД, называемые персональной (настольной), офисной и промышленной (корпоративной). На наш взгляд, эти модели различаются:

- набором решаемых задач и областями применения (структуры этих областей в нашей стране и за рубежом могут значительно различаться);
- применяемыми техническими и программными средствами;
- технологией изготовления программных продуктов и их выходными характеристиками;
- набором знаний – навыков (умений) и даже ментальностью специалистов, работающих в рамках той или иной технологии.

Соответственно, при планировании подготовки специалистов ИТ и разработчиков программного обеспечения (ПО) необходимо четко представлять себе, на какую модель будут ориентированы выпускники той

или иной специальности, и в первую очередь сосредоточить внимание на освоении ими тех средств и методов, выработке у них тех профессиональных подходов, которые выбранной модели свойственны. Вместе с тем, нельзя и замыкаться в рамках единственной модели хотя бы потому, что эти рамки не могут быть определены четко и изменяются со временем. Последние две характеристики моделей ИТ остаются вне сферы внимания большинства авторов, для нас же они представляются не менее важными, чем первые.

Персональная модель наиболее близка к "обывательскому" представлению об АОД. Она предполагает сосредоточение данных и средств их обработки на одном ПК. В развитых странах персональная модель в чистом виде существует только на домашних ПК, не используемых в производственных целях. Любой же ПК, участвующий в бизнесе (даже если это единственный ПК в "карликовой" фирме), является клиентом одного или нескольких серверов и, таким образом, уже не является чисто персональным инструментом. Иначе обстоит дело в постсоветских странах, где имеется значительное количество пользователей, только лишь приступающих к автоматизации обработки своих данных и не располагающих ни значительными вычислительными, ни коммуникационными возможностями. Возможно, на сегодня такие пользователи являются основными заказчиками для отечественных разработчиков ПО и такая ситуация, наверное, сохранится на ближайшие годы.

Технической базой персональной модели являются ПК на основе Intel-совместимых процессоров с операционной средой - MS DOS, Windows 3.x, Windows 9.x. Характерным является применение неспециализированных программных средств разработки - универсальных языков программирования, в некоторых случаях - персональных СУБД.

Технологию изготовления прикладного ПО в персональной модели для условий нашей страны наиболее точно можно охарактеризовать как кустарную. Прежде всего, разработка начинается при отсутствии четко сформулированных и документально зафиксированных спецификаций. Уточнение (также нечеткое) спецификаций происходит уже на этапе внедрения и соответствие им программного изделия (ПИ) достигается методами "наложения заплат". Документально не фиксируется ни ход разработки, ни даже ее конечный результат. В лучшем случае для ПИ изготавливается подобие "Руководства пользователя" и/или экранная подсказка. В любом случае у ПИ остается ряд недокументированных интерфейсов, известных только разработчику. Перенос ПИ на другую установку не поддерживается ни документацией, ни сервисными программными средствами. Как правило, эксплуатация такого ПИ возможна только в условиях, зафиксированных на момент его внедрения. Возникновение нештатных ситуаций или изменение условий требует вмешательства разработчика. Строго говоря, такое ПИ не может считаться программным продуктом. Отчасти такое положение объясняется тем, что

разработчики, не умея или не имея возможности защитить свои права на ПИ, сознательно или интуитивно делают его эксплуатацию зависимой от автора.

Для профессиональной ментальности программистов, работающих в рамках такой модели, характерно отношение к ресурсам вычислительной системы и к обрабатываемым данным, как к своей собственности, с которой они вольны делать "все, что угодно". Название таких программистов - "хакеры" - уже переключалось из профессионального жаргона в общеупотребительный лексикон. Их стиль программирования характеризуется низким уровнем применяемых средств, прежде всего - прямым доступом к ресурсам (системным структурам данных, средствам ввода-вывода и т.п.). В частности этим объясняется тот факт, что "хакерство" процветает в среде MS DOS, ресурсы которой полностью открыты, а также в тех операционных средах, которые хотя и запрещают прямой доступ к ресурсам, но не обеспечивают реальной их защиты. Для этого стиля характерно также недоверие к использованию уже готовых программных модулей и обращение к недокументированным возможностям промышленных программных продуктов (которые, конечно же, используются в нелегальных копиях). К сожалению, средствами массовой культуры создается идеализированный образ хакера, чем обесценивается престиж подлинного профессионала в АОД. Практика же показывает, что внедрение хакеров в неперсональную АОД зачастую приводит к плачевным последствиям.

Офисная модель АОД предполагает обработку данных в локальной сети. В развитых странах эта модель является типовой для небольшого предприятия/офиса, в наших условиях она считается пригодной и для крупного предприятия, например, банка. Число узлов в такой сети может достигать до 100. В наших условиях их количество обычно не превышает 15 -20. Организация АОД в такой сети, как правило, такова, что каждый узел сети является специализированным автоматизированным рабочим местом (АРМ), связанным с ограниченным подмножеством общих данных и с проблемно ориентированными средствами обработки. Данные хранятся либо на локальном диске АРМ, либо на файловом сервере, а средства обработки - на локальном диске. Вместе с тем в отдельных (нетипичных) случаях данные, закрепленные за одним АРМ, могут быть востребованы другим АРМ. Взаимодействие АРМ в таком случае осуществляется методами файлового сервиса. Возможное взаимодействие всей информационной системы с внешними информационными системами осуществляется средствами электронной почты.

Технической базой для офисной модели являются те же обычные ПК в качестве АРМ и более мощный ПК (возможно - сервер на платформе Intel) в качестве файлового сервера. Сетевые операционные системы - Novell Netware с клиентами MS DOS, Windows NT с клиентами

Windows 9.x, по конъюнктурным причинам реже применяется OS/2 Warp, поддерживающая практически любых клиентов.

Инструментальными средствами разработки ПО в данной модели являются прикладные инструментальные среды для ПК (MS Office, Delphi, Visual FoxPro, etc.). Такие среды обладают более или менее развитыми собственными средствами программирования, как правило, достаточными для решения многих задач, возникающих в офисной модели.

Основным содержанием задач офисной модели является хранение и выборка информации, т.е., ведение баз данных. Общеизвестно, что разработка базы данных должна начинаться с создания инфологической и концептуальной схем, которые определяют и интегрируют все прочие компоненты разработки. На практике же в наших условиях строгая концептуальная схема отсутствует даже для локальной (связанной с одним АРМ) базы данных, не говоря уже обо всей системе в целом. Документирование разработок также оставляет желать лучшего, ограничиваясь инструкцией оператора и описаниями таблиц базы данных.

Для ментальности программистов, работающих в рамках офисной модели, характерно чрезмерное доверие к возможностям выбранных ими инструментальных средств. Ими часто декларируется, что все, что необходимо для решения любой задачи, имеется среди базовых возможностей данного инструмента (например, MS Access), а все, чего нет среди этих возможностей (или есть, но не очень легко обеспечивается) объявляется просто лишним. Таким образом, в разряд "лишних" попадают такие качества информационной системы, как обеспечение многопользовательского доступа к единственной копии данных, независимость данных от средств их обработки, "тонкое" распределение прав доступа к данным, единое администрирование системы и многое другое. Это приводит в итоге к дублированию в сети данных и программ, ограничению доступности данных, необходимости отдельного системного обслуживания каждого АРМ, затруднениям при развитии связей с внешними информационными системами. Еще одной особенностью психологии программистов является увлечение графическим интерфейсом. Сами инструментальные средства очень интенсивно используют возможности визуализации и по их подобию программист строит интерфейс прикладного продукта обычно не слишком заботясь об обосновании его эргономичности. Экран конечного пользователя может выглядеть как красивая и эффективная картинка, но быть перегруженным органами управления, данными и просто "арабесками", что затрудняет для непрофессионала его освоение и работу в нем. К тому же, многие приложения такого рода должны выполняться в той же среде, в которой они разрабатывались, поэтому у конечного пользователя остается доступ и к инструментальным средствам, и его неправильные действия - намеренные или ошибочные - могут привести к порче данных и программ.

Специалисты по программированию в офисной модели, безусловно, необходимы и востребованы, но они должны четко представлять себе ее возможности и ограничения. От таких специалистов требуется свободное владение возможностями целого ряда соответствующих инструментальных средств, в том числе и тех, которые "не лежат на поверхности". От них требуется умение создавать проект программного продукта и следовать ему в процессе реализации ПИ. От них требуется понимание и выполнение эргономических требований к рабочему месту конечного пользователя. От них требуется соблюдать стандарты де-юре и де-факто, обеспечивающие взаимодействие как отдельных АРМ, так и целых информационных систем. От них, наконец, требуется умение распознать момент, когда количественное разрастание и качественное развитие информационной системы переведут ее в иную модель АОД.

Промышленная модель АОД названа так по ряду причин. Во-первых, в такой модели данные обрабатываются в промышленных масштабах, т.е. в больших и сверхбольших объемах. Соответственно повышаются и требования к качеству, надежности и защищенности их хранения и обработки. Во-вторых, конечный продукт разработки ПО обладает признаками промышленного продукта: он может функционировать отдельно от своего автора, может продаваться и покупаться. При этом интересы и автора, и продавца, и потребителя программного продукта должны быть защищены. В-третьих, производство ПО также обладает признаками промышленного производства, а именно, плановостью и относительной независимостью процесса производства от отдельных его участников.

Технические средства промышленной АОД отличаются большим разнообразием. Систему промышленной АОД можно начинать строить на базе технических средств офисной обработки. При увеличении числа АРМ в системе экономически выгодным становится увеличение мощности сервера при снижении требований к локальным ресурсам АРМ. Современный подход к построению таких систем базируется на 3х-уровневой архитектуре клиент/сервер (сервер баз данных - сервер приложений - клиенты). В качестве серверов используются многопроцессорные серверы или кластеры на базе Intel, серверы на базе RISC-процессоров, мейнфреймы. АРМ же могут быть представляться как обычными ПК, так и сетевыми компьютерами, неинтеллектуальными терминалами или мобильными устройствами ("тонкие" клиенты). Программные средства промышленной АОД также отличаются большим разнообразием. Если в ПО на базе ПК практически бесспорна монополия фирмы Microsoft, то в производстве ПО промышленных систем с равным успехом действуют и взаимодействуют такие фирмы, как IBM, Sun, Oracle, Inprise и многие другие. Наряду с операционными системами огромную роль в таких системах играет промежуточное программное обеспечение (middleware): серверы приложений (IBM WebSphere, Oracle

Application Server, Lotus Domino, etc.), СУБД (Oracle, IBM DB2, MS SQLServer, etc.), менеджеры транзакций (IBM CICS, Transare Encina, MS Transaction Server, etc.), системы передачи сообщений (IBM MQSeries, MS MQServer, etc.), системы управления распределенными ресурсами (Tivoli). Каждый из производителей такого ПО представляет собственные инструментальные средства для прикладных разработок в его среде, но все эти средства имеют некоторые общие концепции и подходы. К таковым можно отнести компонентный подход, который при создании приложения заменяет программирование сборкой из готовых компонент. К ним относятся средства сборки, использующие концепции визуального проектирования и программирования (ERWin, Rational Rose, VisualAge). К ним относятся также предустановленные "каркасы" (framework), содержащие богатейшие наборы компонент для определенных классов прикладных задач (например, IBM SanFrancisco). Для инструментария большинства фирм характерным является возможность построения гетерогенных приложений, включающих в себя изделия разных производителей. Основными средствами интеграции являются язык Java, языки Web-представления (HTML, XML), технологии EJB, CORBA и DCOM.

Ментальность программистов, работающих в промышленной модели, характеризуется прежде всего высоким чувством ответственности за конечный продукт. В приоритетах оценок ПИ эффективность (а иногда даже и эффективность) найденных решений уступает надежности и функциональности. Оригинальные решения не поощряются, если они подменяют апробированные решения, имеющиеся в "каркасах". Программист не может считать свой продукт полноценным, если он не укомплектован исчерпывающей документацией. Определяющей целью разработчиков ПО в промышленной модели является создание программного продукта с заданными свойствами в заданные сроки с минимизацией затрат на его создание и последующее сопровождение.

Разумеется, при проектировании промышленного программного изделия могут использоваться только лицензионные средства, компоненты, базовое ПО. Вместе с тем, сам результат такого проектирования является промышленным продуктом и должен быть защищен как от несанкционированного использования в техническом (программном) отношении, так и от неправомерного использования в правовом отношении четким определением авторских прав на него. Только в этом случае возможна профессиональная коммерциализация программного продукта и превращение его в объект купли/продажи.

Для данной модели требуется широкий спектр специалистов ИТ, ориентированных как на технические и программные средства, так и на вопросы защиты и коммерциализации продукта. Однако, даже преимущественно технические специалисты наряду с владением инструментарием разработки, методологией и технологией проектирования должны

владеть основами инновационного менеджмента, правовой защиты промышленной собственности и авторского права, так как эти вопросы могут оказывать существенное влияние на содержание и ход разработки ПИ.

В трехуровневой архитектуре клиент/сервер заложена возможность динамического соединения клиента с несколькими серверами приложений, а сервера приложений - с несколькими серверами баз данных. В итоге это ведет к созданию единого глобального пространства информационных и вычислительных ресурсов [2]. Как следствие этого, подходы, принятые в промышленной модели становятся доминирующими, вычисления становятся сервер-центрическими, функция клиента ограничивается только презентацией информации конечному пользователю. ПК в этой роли представляют собой только одну из возможных альтернатив наряду с мобильными и тонкими клиентами. Однако, в отличие от "персональной революции", процесс перехода к новой архитектуре носит эволюционный характер. Это означает, что персональная и офисная модель еще долго будут сосуществовать с промышленной и специалисты в этих моделях будут востребованы.

Таким образом, при составлении программ и учебных планов по подготовке специалистов ИТ мы считаем необходимым для преподавательского коллектива прежде всего определиться в отношении той модели, к которой будут адресоваться выпускники. В зависимости от выбранной модели формируется номенклатура, объем и содержание читаемых курсов, а также техническое и программное обеспечение лабораторной базы.

Изложенная точка зрения авторов нашла отражение в концепции развития специализации "Инжиниринг и коммерциализация программного обеспечения", по которой с 2001 г. осуществляется подготовка специалистов на кафедре информатики и интеллектуальной собственности Национального технического университета "Харьковский политехнический институт".

## ЛИТЕРАТУРА

1. Bolthouse D. Exploring IBM Client/Server Computing. – MAXIMUM PRESS, 1996. – 445 p.
2. Технологии IBM для электронного бизнеса. – IBM East Europe / Asia, 2001. – 240 с.

*Поступила в редколлегию 15.10.2001*