

УДК 681.3

ИСПОЛЬЗОВАНИЕ WEB-ТЕХНОЛОГИЙ ДЛЯ АВТОМАТИЗАЦИИ УПРАВЛЕНИЯ РЕСУРСАМИ БАНКА

Михальченко В.П.

Показаны новые возможности автоматизированного управления ресурсами кредитного учреждения на основе использования WWW-сервиса глобальной сети «INTERNET». Предлагается использование технологии ASP для экспресс-оценки финансового состояния банка.

Банковские учреждения в Украине в настоящее время относятся к одним из наиболее перспективных объектов автоматизации, динамично и эффективно внедряющих прогрессивные принципы управления, информационные технологии и программно-технические средства практически во всех областях своей деятельности. С одной стороны, это обусловлено довольно жесткой конкуренцией на рынке банковских услуг – сегодня клиент выбирает только тот банк, который обеспечивает быстрое и качественное обслуживание его платежей (автоматизированные системы «Клиент-Банк»), достоверное информирование о состоянии его счетов и другой финансовой информации в реальном масштабе времени (системы «Банк-Пейджер», «Теле-Банк», «Банк-Факс»), оперативный и объективный мониторинг его кредитных проектов и др., гарантируя при этом высокую степень конфиденциальности и безопасности информации. Аналогичные критерии оказывают немалое влияние и на выбор банков-партнеров на рынке межбанковских ресурсов.

С другой стороны, решительная политика Национального банка Украины (НБУ) в области автоматизации, в частности, перевод в 1995 г. всей банковской системы исключительно на электронные платежи (впервые в странах СНГ), а также внедрение в 1998 г. жесткой системы финансовой и статистической отчетности по электронной почте, просто заставили все банковские учреждения Украины активно внедрять системы автоматизации банков (САБ) и средства телекоммуникаций

К настоящему моменту времени практически все основные конкуренты на рынке САБ предлагают программные продукты, в значительной степени автоматизирующие большинство направлений операционной работы банков, включая расчетно-кассовое обслуживание, кредитование, депозитные и дилинговые операции и пр. (OLTP - on-line transaction processing) Главным направлением дальнейшего развития САБ сегодня стала разработка и внедрение аналитических подсистем (OLAP - on-line analyzing processing) по выработке решений в области наиболее

эффективного управления ресурсами [1]. При этом предпринимаются многочисленные попытки привлечь для этого методы статистического и имитационного моделирования, теории искусственного интеллекта, исследования операций, нейросетевых технологий и пр.

Для оптимального управления кредитными рисками банки должны тщательно выбирать партнеров на межбанковском рынке ресурсов, адекватно оценивая их финансовое состояние и в соответствии с этим устанавливая лимиты на разные виды сделок. Как правило, каждый банк разрабатывает собственную методику экспресс-анализа балансов банков-контрагентов и определения рейтинговых оценок. Ниже приводится один из возможных вариантов этой методики.

Состояние деятельности банка характеризуют:

- виды активных операций (направления размещения средств банка) и их структура;
- виды пассивных операций (источников средств банка) и их структура;
- показатели достаточности капитала и платежеспособности;
- показатели ликвидности баланса (способности своевременно исполнить свои денежные обязательства);
- прибыльность банка.

Для определения финансового состояния банка предлагается использовать Z-статистику, построенную на использовании модели Альтмана [2]

$$Z = \sum_{i=1}^4 k_i b_i \quad (1)$$

где b_i - регрессионные коэффициенты, полученные в результате анализа накопленных данных предыдущих балансов данного банка.

k_i - оценки, рассчитываемые по формулам

$$k_1 = \frac{D_0}{A} \quad - \text{оценка прибыльности активов}, \quad (2)$$

где D_0 - чистая прибыль (разность между счетами 6-го и 7-го класса за минусом начисленных доходов и расходов, учитываемых на счетах 2-го класса), A - совокупные активы;

$$k_2 = \frac{S_1}{S_2} \quad - \text{коэффициент финансовой независимости}, \quad (3)$$

где S_1 - собственные средства, S_2 - привлеченные средства,

$$k_3 = \frac{K}{A}, \quad (4)$$

где K - основной капитал (сумма счетов 5-го класса);

$$k_4 = \frac{D_b}{A}, \quad (5)$$

где D_b - брутто-доход (доход отраженный в консолидированном балансе банка).

Если полученное значение Z превышает 2,99, то банк является стабильным, если его значение меньше 1,81 - риск выдачи кредита очень высок.

Рейтинг банка для определения лимита межбанковского кредита устанавливается путем анализа результатов расчета и данных формы №611 "Отчет о выполнении обязательных нормативов":

$$r = \begin{cases} Z + 1 & , \text{ если банк выполняет установленные НБУ экономические} \\ & \text{нормативы;} \\ Z & , \text{ в противном случае.} \end{cases}$$

Лимит кредитования рассчитывается в процентах к сумме основного капитала банка (сумме счетов 5-го класса баланса) по формуле

$$L = rK, \quad (6)$$

после чего распределяется по срокам выдачи.

Важным аспектом успешного внедрения OLAP-технологий в реальную эксплуатацию является обеспечение высокой степени достоверности и актуальности исходных данных. Для «внутренних» источников информации из разных подсистем САБ это реализуется без особых проблем, особенно в случае САБ с архитектурой «клиент-сервер». Для работы с «внешними» источниками нужны надежные, быстрые и желательно недорогие каналы связи.

Развитие глобальной сети «INTERNET», объединившей миллионы компьютеров в единое информационное пространство, появление простых способов доступа в сеть, а также широкая распространенность удобных программ, поддерживающих различные сервисы сети, давно привлекали внимание разработчиков САБ в их поисках подходящих средств транспорта информации между банком и клиентом, головным банком и филиалами, банками-контрагентами и др. Так, например, значительное количество разработок систем «Клиент-Банк» использует «INTERNET», широко распространен обмен информацией посредством электронной почты E-mail между банками, филиалами и отделениями, многие банки используют WWW-сайты для рекламы своих услуг, информирования своих клиентов и партнеров об услугах, текущих котировках и пр.

Одним из важнейших условий успешной деятельности любого банка, активно работающего в сфере кредитования, является возможность точной оценки финансового состояния потенциальных партнеров. Весьма актуален был бы автоматизированный мониторинг кредитных проектов предприятий-клиентов по результатам анализа их финансовой отчетности, однако появление таких OLAP-систем пока сдерживается необходимостью разработки и внедрения специальных стандартов для предоставления заемщиками своих отчетов (бухгалтерских балансов и т.п.) в виде, пригодном для автоматизированной обработки. В лучшем случае работник кредитного отдела сегодня имеет для автоматизации финансового анализа офисное приложение, позволяющее ему «вручную» ввести исходные данные из «бумажных» отчетов потенциальных заемщиков.

Совсем иная ситуация в области межбанковского кредитования (МБК). Благодаря упомянутому выше внедрению автоматизированной системы отчетности НБУ каждое банковское учреждение формирует и отправляет электронной почтой в адрес соответствующего подразделения НБУ строго регламентированный набор отчетных файлов (более десятка файлов ежедневно, более трех десятков ежемесячно, около десятка ежеквартально) с жестко стандартизированными форматами данных каждого, причем возможные нарушения регламента немедленно влекут за собой штрафные санкции. Поэтому есть огромный комплект унифицированных электронных отчетов по всем направлениям работы каждого банка, который безо всяких дополнительных усилий можно использовать для пересылки предполагаемым партнерам по МБК с целью всесторонней оценки финансового состояния данного банка.

Сегодня в Украине широко распространена следующая практика установления лимитов банкам-контрагентам по МБК. Один раз в месяц (обычно до 10-го числа) все участники рынка МБК обмениваются файлами по форме #1 «Данные об остатках на балансовых счетах», используя в большинстве случаев электронную почту НБУ или E-mail.

Используя приложение «АРМ «Статотчетность», разработанное НБУ, каждый банк может по этим файлам оценить отдельные показатели финансового состояния потенциальных партнеров. Некоторые банки имеют специальные приложения, которые по данным этих файлов автоматически рассчитывают лимиты на сделки по МБК по методикам, аналогичным (1)-(6).

Значительно более удобным и современным представляется следующий подход основанный на использовании WEB-технологии «INTERNET». Банк размещает на своем WWW-сайте активную WEB-страницу, интерфейс которой позволяет каждому банку, желающему сотрудничать на рынке МБК, обычной программой просмотра загрузить свой (или любой другой) отчетный файл #1 с балансом, и оперативно

получить результат его анализа и размеры лимитов, которые могут быть предоставлены данному банку для сделок по МБК.

Для реализации этого подхода разрабатывается WEB-система с трехъярусной моделью организации приложений. Суть трехъярусной модели – в разделении уровня клиентских представлений (пользовательского интерфейса), уровня бизнес-процессов (собственно содержательной части задачи) и уровня управления данными:

“Клиент $\Leftrightarrow$ WEB-сервер + сервер приложений $\Leftrightarrow$ сервер базы данных”.

В качестве клиента WEB-системы выступают программы просмотра. Таким образом, это типичный пример архитектуры с “тонким клиентом”(thin). “Толстый клиент” (fat) сам реализует основные бизнес-процессы, используя сервер для организации доступа к базам данных и выполнения самых общих задач. На “тонкого клиента” возлагается только функция поддержки пользовательского интерфейса. Вся осмысленная работа выполняется на сервере приложений.

Сервер приложения отвечает за выполнение бизнес-процессов на сервере, т.е. в описываемой системе за расчет лимитов по принятым файлам балансов, накоплению данных в базе для их регрессионного анализа, обновления текущих котировок и т.п. Это могут быть выполняемые на WEB-сервере CGI-программы, модули ISAPI, а также технология Active Server Pages (ASP) – открытое окружение объединяющее страницы HTML, серверные компоненты ActiveX и ActiveX Scripting Engine для создания динамических интерактивных серверных приложений[3].

Традиционная схема построения подобных систем изображена на рисунке 1.

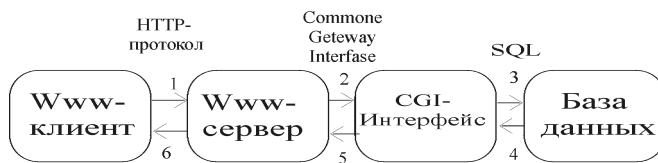


Рисунок 1 –Архитектура WEB-системы

Более современный подход основан на Microsoft Active Server Pages (ASP) — это среда для разработки сценариев на стороне сервера, которую можно использовать для создания интерактивных WEB-страниц и построения мощных WEB-приложений. Когда сервер получает запрос на файл ASP, он обрабатывает сценарий на стороне сервера, содержащийся в файле, и формирует WEB-страницу, которая передается в обозреватель. Кроме сценариев на стороне сервера файлы ASP могут содержать HTML

(включая связанные сценарии на стороне клиента), а также вызовы компонентов COM, которые выполняют множество задач, например, соединение с базой данных или обработка бизнес-логики.

Создание серверных сценариев с помощью ASP — быстрый и простой способ перейти к разработке сложных коммерческих WEB-приложений. ASP также предлагает универсальное решение задачи хранения в базе данных сведений, которые представлены HTML-формой, настройки WEB-узлов с учетом особенностей посетителей и использования разнообразных возможностей обозревателей при работе с HTML-страницами. Например, раньше создание обычного приложения Common Gateway Interface (CGI) для обработки получаемых от пользователя данных на WEB-сервере требовало знания языка PERL или C. Располагая возможностями ASP, можно получать данные HTML-форм и передавать их в базу данных с помощью простых серверных сценариев, которые могут включаться непосредственно в HTML-документы.

В ASP отсутствует ориентация на конкретный язык программирования, поэтому знакомства с любым языком сценариев (VBScript, JScript или PERL) достаточно для того, чтобы работать с Active Server Pages. Более того, на страницах ASP допускается использование любого языка сценариев, для которого был установлен COM-совместимый обработчик сценариев. В состав ASP входят обработчики сценариев VBScript и JScript, но имеется дополнительная возможность установки обработчиков для языков PERL, REXX и Python, которые могут быть получены от независимых разработчиков.

Профессиональным разработчикам, программирующим на языках Visual Basic, C++, Java и др., ASP обеспечивает гибкие возможности быстрого создания законченных WEB-приложений. Помимо добавления сценариев и создания удобного HTML-интерфейса приложения имеется также возможность самостоятельной сборки компонентов COM. Необходимые правила расчетов могут быть реализованы в универсальных модулях, которые могут многократно вызываться из сценариев, а также из других компонентов программ.

Выполнение серверного сценария начинается после поступления от WEB-обозревателя запроса на файл «.asp». После этого WEB-сервер вызывает ASP для обработки указанного в запросе файла. Обработка этого файла происходит последовательно от начала и до конца, при этом выполняются все содержащиеся в нем команды сценария, после чего файл отправляется на обозреватель в виде WEB-страницы.

Поскольку сценарии выполняются на сервере, а не на локальном компьютере, WEB-сервер выполняет и всю необходимую работу по созданию WEB-страниц, которые будут отправлены в обозреватель. Простое копирование серверных сценариев невозможно, поскольку в

обозреватель передаются только результаты их выполнения. Пользователи могут просматривать WEB-страницы, но не могут видеть текст сценария, с помощью которого эти страницы создавались. Таким образом, сама методика расчета лимитов по МБК может остаться закрытой для банков-контрагентов

Языки программирования, например Visual Basic, C++ и Java, обеспечивают доступ на нижнем уровне к ресурсам компьютера. Языки написания сценариев, в свою очередь, используются для создания программ ограниченных возможностей, называемых сценариями, которые выполняют функции WEB-узла на WEB-сервере или в обозревателе. В отличие от сложных языков программирования языки написания сценариев интерпретируются инструкции последовательно выполняются промежуточной программой, называемой интерпретатором команд. Хотя интерпретация уменьшает эффективность выполнения, языки написания сценариев просты для изучения и обеспечивают большие возможности. Сценарии могут быть встроены в HTML-страницы для форматирования содержимого или могут реализовывать компоненты COM, заключающие в себе бизнес-логику.

Компонент COM представляет собой повторно используемый блок инструкций. Компонент предназначен для построения программ и обеспечивает решение некоторой задачи или набора задач. Чтобы создать WEB-приложение, компоненты объединяют друг с другом, в том числе посредством сетей. Компоненты COM обеспечивают решение общих задач, избавляя пользователей от необходимости самим создавать соответствующие программы. Например, компонент для чтения банковских новостей можно использовать для отображения на WEB-странице последних банковских котировок

В комплект поставки ASP входят готовые основные компоненты. Например, с помощью компонентов ADO (объектов данных ActiveX) нетрудно добавить к имеющимся WEB-страницам возможности подключения к базам данных. Компоненты являются лучшим средством реализации бизнес-логики в виде защищенных повторно используемых модулей. Например, с помощью компонента можно обеспечить проверку банковских номеров МФО посредством вызова компонента в сценарии, обрабатывающем файлы балансов. Поскольку такая проверка выполняется отдельно от остальных процессов, соответствующий компонент нетрудно обновить, если в процесс проверки номеров МФО будут внесены изменения, при этом процесс обработки файлов балансов останется без изменений. Кроме того, компоненты COM можно повторно использовать в других сценариях и приложениях. После установки компонента на WEB-сервере данный компонент может быть вызван из сценария ASP на стороне сервера, из расширения ISAPI, из другого компонента на сервере, а также

из программы, написанной на другом языке, совместимом со стандартом COM.

Объекты ADO представляют собой простую, но мощную технологию предоставления доступа к базам данных с WEB-страниц. Объекты ADO позволяют писать компактные и расширяемые сценарии для подключения к совместимым с интерфейсом OLE DB источникам данных, например, базам данных, электронным таблицам, последовательным файлам данных или каталогам электронной почты. OLE DB представляет собой программный интерфейс системного уровня, который предоставляет стандартный набор интерфейсов COM для обеспечения работы системы управления данными. Это необходимо для накопления данных обрабатываемых файлов баланса для последующего регрессионного анализа (1).

Библиография

1 Куванина Д.П. Системный подход к лимитной политике и управлению банковскими рисками / Д.П.Куванина // Системный анализ в проектировании и управлении: Материалы докл. науч.-практ. конф., СПб, 20-22июня 2001г. — СПб, 2001. — С. 141 – 142

2 Егорова Н.Е. Математические методы финансового анализа банковской деятельности / Н.Е.Егорова, А.М.Смулов // Аудит и финансовый анализ. — 1998. — №2. — С. 75 – 146

3 Хилайер С. Программирование Active Server Pages / С. Хилайер, Д. Мизик. — М.: Изд-во «Русская редакция», 2000. — 320 с.

Поступила в редакцию 19.12.2001 г.