УДК 004.415.2.052.03

С.Е. Голиков

Севастопольский институт банковского дела, Севастополь, Украина

ДИНАМИЧЕСКИ НАСТРАИВАЕМАЯ ИНФРАСТРУКТУРА ИНФОРМАЦИОННОЙ СИСТЕМЫ БАНКА

В статье рассмотрен механизм минимизации информационно-технологической составляющей операционных рисков, возникающей вследствие использования систем автоматизации в банковских учреждениях Украины. Показано, что операционные риски являются основной причиной потерь значительных денежных сумм финансовых организаций. Рассмотрены инфраструктурные решения современных систем автоматизации банковской деятельности, выявлены их узкие места. Предложен механизм динамически настраиваемой инфраструктуры информационной системы, позволяющий существенно повысить ее отказоустойчивость и доступность. Показана возможность использования разработанного механизма для снижения функциональной составляющей операционного риска. Рассмотренные решения нашли применение в конкретных проектах.

Ключевые слова: *операционный риск, система автоматизации банка, отказоустойчивость, резервирование, доступность, сервер приложений, трехуровневая архитектура.*

Введение

В последние годы кризисные явления в мировой экономике заставляют кредитные учреждения в первую очередь направлять усилия на удовлетворение потребностей клиентов, что, в свою очередь, требует интенсивного использования информационных систем. Информационные системы создают благоприятные условия для привлечения новых клиентов и позволяют поддерживать конкурентные преимущества для уже существующей клиентской базы. Результаты исследований [1] говорят о том, что доля операционного риска в причинах потерь финансовых организаций значительных денежных сумм постоянно увеличивается.

Постановка задачи. Таким образом, защита информационных систем от влияния операционных рисков выходит на первый план. В связи с этим, разработка и внедрение механизмов, позволяющих минимизировать операционные риски финансовых учреждений, является актуальным направлением для исследований в условиях глобализации финансовых слуг наряду с возрастающей сложностью применяемых в них информационных технологий.

Основная часть

Базельский комитет по банковскому надзору в своем документе под названием «New Basle Capital Accord» [6] дает определение операционному риску как «риску потерь вследствие неадекватных или неудачных внутренних процессов, людей и систем или внешних событий». НБУ дает определение операционному риску следующим образом: в «Положении об организации операционной деятельности в банках Украины»[7] под операционным риском понимается «риск, который связан с нарушением банковских правил и/или систем контроля за обработкой, проведением операций, документацией, который возникает как вследствие внешних причин, так и из-за ошибок работников банка». В Методических указаниях по инспектированию банков «Система оценки рисков» операционный риск определяется как «потенциальный риск для существования банка, который возникает из-за недостатков корпоративного управления, системы внутреннего контроля или неадекватности информационных технологий и процессов обработки информации с точки зрения управляемости, универсальности, надежности и непрерывности работы» [8].

Операционные риски включают риски человеческого фактора (ошибки, внешнее и внутреннее воровство, болезни и т.п.), риски технологий (поломки оборудования, ошибки в программном обеспечении, системные сбои в работе и т.п.), риски внешних факторов (стихийные бедствия, катастрофы и пр.) [4]. В отличие от других видов рисков, источник операционных рисков чаще всего лежит внутри системы. Таким образом, устранив порождающие причины, можно снизить вероятность возникновения операционных рисков. В виду того, что реализация банковских продуктов (оказание банковских услуг) в настоящее время происходит непосредственно в рамках автоматической банковской системы, банковский операционный риск имеет смысл рассматривать исключительно в рамках АБС [2].

Самым слабым звеном отечественных банков НБУ считает информационно-техническую составляющую операционных рисков. Операционно-технологический риск возникает из-за неадекватности информационных технологий и процессов обра-

ботки информации, в том числе вследствие неадекватности стратегии, политики и использования информационных технологий. К другим аспектам операционно-технологического риска относятся вероятность непредвиденных событий, например, пожара или стихийного бедствия [3]. Поэтому регулятором утверждены стандарты информационной безопасности для банков [12] и разработаны методические рекомендации по их внедрению, в соответствии с которыми планируется проведение проверок банков в этом направлении.

В рамках данной статьи рассматривается технологическая составляющая операционного риска, а именно архитектурная устойчивость систем автоматизации банков по отношению к сбоям. Под архитектурной устойчивостью к сбоям понимается создание такой архитектуры приложения, которая отвечает всем техническим и операционным требованиям и обеспечивает оптимальные производительность, доступность, надежность и безопасность [5]. Отказоустойчивость является одним из важнейших требований к информационной системе. Целью данной работы является проектирование элементов архитектуры системы автоматизации банков, позволяющей существенно снизить вероятность возникновения инцидентов, вследствие сбоев и отказов информационных систем. Правильно спроектированная архитектура существенно снижает операционные риски, связанные с созданием и функционированием программной системы. Архитектура программного обеспечения систем автоматизации банков представляет собой комплекс, состоящий из набора компонентов, выполняющих определенную функцию или набор функций. Современные банковские системы представляют собой совокупность различных архитектурных стилей. Наиболее часто используется архитектурная парадигма – клиент-сервер (N-уровневая) с элементами компонентной и многоуровневой. Клиент-серверная (N-уровневая) архитектура позволяет разместить инфраструктурные элементы на разные физические компьютеры, повышая отказоустойчивость всей системы, обеспечивая централизованный доступ к данным и простоту обслуживания, а компонентная архитектура упрощает функциональное наращивание путем повторного использования логических компонентов. Многослойность позволяет разделить логику представления от бизнес-логики и логики доступа к данным, что обусловлено требованиями банковской безопасности. В качестве модели взаимодействия на уровне представления в основном применяется шаблон представления с разделением (модель – представление – контроллер). В отечественных банковских системах, как правило, применяется трехуровневая архитектура, в которой программные сегменты физически размещены на разных компьютерах. Преимуществами наличия промежуточного слоя программного обеспечения являются [9]:

- снижение сложности создания информационной системы;
- прозрачная маршрутизация запросов;
- балансировка загрузки вычислений, что повышает доступность данных;
- простота настройки и переконфигурации;
- масштабируемость.

Для связи между элементами программной инфраструктуры используются сообщения. Характеристиками такой архитектуры являются функциональная декомпозиция программной системы, распределенное развертывание, что обеспечивает повышенную масштабируемость, доступность, управляемость и эффективность использования ресурсов. Каждый уровень функционально изолирован от других, кроме тех, с которыми он непосредственно соседствует. Бизнес-слой в данной архитектуре развернут на сервере приложений, слой представления – на клиентских компьютерах, а сами данные – в базе данных, что повышает конфиденциальность и безопасность. Размещение бизнес-логики на отдельном сервере гарантирует доступность изменений для всех пользователей системы, изменения в настройках производятся централизованно, ответственность за аутентификацию пользователей переносится с потенциально небезопасного клиентского уровня на уровень сервера приложений, скрывая уровень базы данных. Наиболее известными промышленными серверами приложений являются WebSphere Application Server, Oracle Application Server, Glassfish. Основные недостатки подобных серверов:

- высокая стоимость;
- сложность администрирования и управления;
- низкое быстродействие.

Главной проблемой централизованной обработки данных является отказоустойчивость сервера приложений, так как в случае выхода его из строя клиенты не смогут получить доступ к приложению. В современных банковских автоматизированных системах, применяемых в коммерческих банках Украины, данной проблеме не уделено должное внимание. Отказоустойчивость достигается либо средствами операционной системы, либо средствами СУБД, либо применением аппаратных решений. Однако применение вышеперечисленных способов требует высокой квалификации персонала и значительных денежных вложений. Намного более простым способом представляется достижение решения данной проблемы за счет резервирования элементов сервера приложений, предусматривающего перераспределение запросов на исправные компоненты платформы. Одним из основных методов повыше-

ния отказоустойчивости является резервирование, за счет чего можно безгранично повышать надежность и отказоустойчивость инфраструктуры[10].

Предлагается использовать резервирование замещением, при котором резервный сервер приложений включается в работу автоматически при перенаправлении на него запросов от клиентских приложений. Вероятность безотказной работы резервированной системы рассчитывается по формуле:

$$P(t)_p = 1 - \left[1 - P(t)\right]^{m+1}, \quad (1)$$

где $P(t)_p$ – вероятность безотказной работы резервированной системы. Если $P(t) = e_p^{-\lambda t}$ – вероятность безотказной работы нерезервированной системы при экспоненциальном распределении надежности, m – кратность резервирования, то

$$T_{cp\,p} = T_{cp} \sum_{i=0}^{m} \frac{1}{(i-1)} = T_{cp}\left(1 + \frac{1}{2} + ... + \frac{1}{m+1}\right), \quad (2)$$

где $T_{cp\,p}$ – средняя наработка на отказ, T_{cp} – средняя наработка на отказ нерезервированной системы.

Для m = 1 получаем

$$P(t)_p = 1 - \left[1 - P(t)\right]^2, \quad (3)$$

$$T_{cp\,p} = 1.5 T_{cp}. \quad (4)$$

Из вышеприведенных формул видно, при простом дублировании средняя наработка на отказ увеличивается в 1.5 раза. В работе [11] показано, что в неоднородных гетерогенных средах среднее значение надежности информационной системы можно выразить в виде:

$$E(N(P)) = \sum_{i=0}^{m} \left(Ni(P) \times Pr[X = i]\right), \quad (5)$$

где $N_0(P)$ – значение надежности при безотказной работе сервера, Ni(P) ($1 \le i \ge m$) значение надежности при возникновения отказа p_i сервера. $Pr[X = i]$ определяет отказоустойчивость сервера, X определяет состояние системы. X=0 означает, что система работает безотказно, а X=i ($1 \le i \ge m$) показывает, что сервер p_i постоянно сталкивается с отказом.

Суть предлагаемого решения состоит в применении резервирования для сервера приложений, осуществляющего функции обработки бизнес-логики и доступа к данным, хранящихся в базе данных. При выходе из строя основного сервера приложений клиентское приложение автоматически подключается к одному из резервных серверов приложений, обеспечивая доступность данных. Даже в случае отсутствия доступного сервера приложений система останется работоспособной.

Инфраструктура системы перестроится в 2.5 уровневую архитектуру, а обработка бизнес-логики и доступ к данным будет осуществляться при помощи стандартных средств используемой СУБД (хранимые процедуры и триггеры).

Для автоматического дублирования сервера приложений используются механизмы:

- автоматического переключения потоков данных на резервный сервер в случае отказа основного;
- автоматического определения статуса сервера "*основной*" или "*резервный*" при старте системы и автоматическое разрешение конфликтов статуса при восстановлении основного сервера после сбоя;
- протоколирования всех сбоев и переключений на резервные серверы.

На рис. 1 показана предлагаемая топология информационной системы банка, обеспечивающая высокую доступность. Сеть делится на несколько сегментов, в каждом из которых размещается ферма серверов приложений.

Таким образом, устраняется единственная точка отказа – сервер – приложений.

В случае отказа одного из серверов приложений состояние и данные будут восстановлены на резервном сервере. В случае выхода из строя всех серверов приложение будет работать напрямую с базой данной. База данных также вынесена за пределы границ фермы серверов приложений.

На рис. 2 изображена структурная схема решения, состоящая из серверной и клиентской части. Сообщения пересылаются асинхронно, что позволяет устранить неблагоприятное влияние на удобство использования и время отклика при возникновении задержек и обрывов связи. Очереди являются общими ресурсами 2-х потоков, поэтому обращения к ним должны быть размещены в критических секциях. Для синхронизации потоков используется алгоритм Питерсона. Объект соединения клиентской части платформы обеспечивает первичную обработку пересылаемой и поступающей информации (выделяет из общего потока системные сообщения), управляет очередями сообщений. Управляющий поток обеспечивает интерфейс работы с удаленным сервером. При наличии сообщения, которое необходимо отправить, объект управляющий поток помещает его в очередь отправляемых сообщений и отсылает через объект соединения. Следующее сообщение посылается только после получения уведомления о готовности объекта соединения к передаче следующего сообщения.

Распределитель сообщений серверной части платформы предназначен для передачи сообщений от одного удаленного объекта другим, проверки прав подключаемых пользователей, создания новых удаленных объектов соединения. Кроме того, распределитель сообщения взаимодействует с объектами базы данных для проверки прав подключившихся пользователей. С целью исключения отказов все отправляемые сообщения сначала помещаются в очередь сообщений, а затем обслуживаются. Кроме того, распределитель сообщений следит за удалением объектов отсоединившихся клиентов.

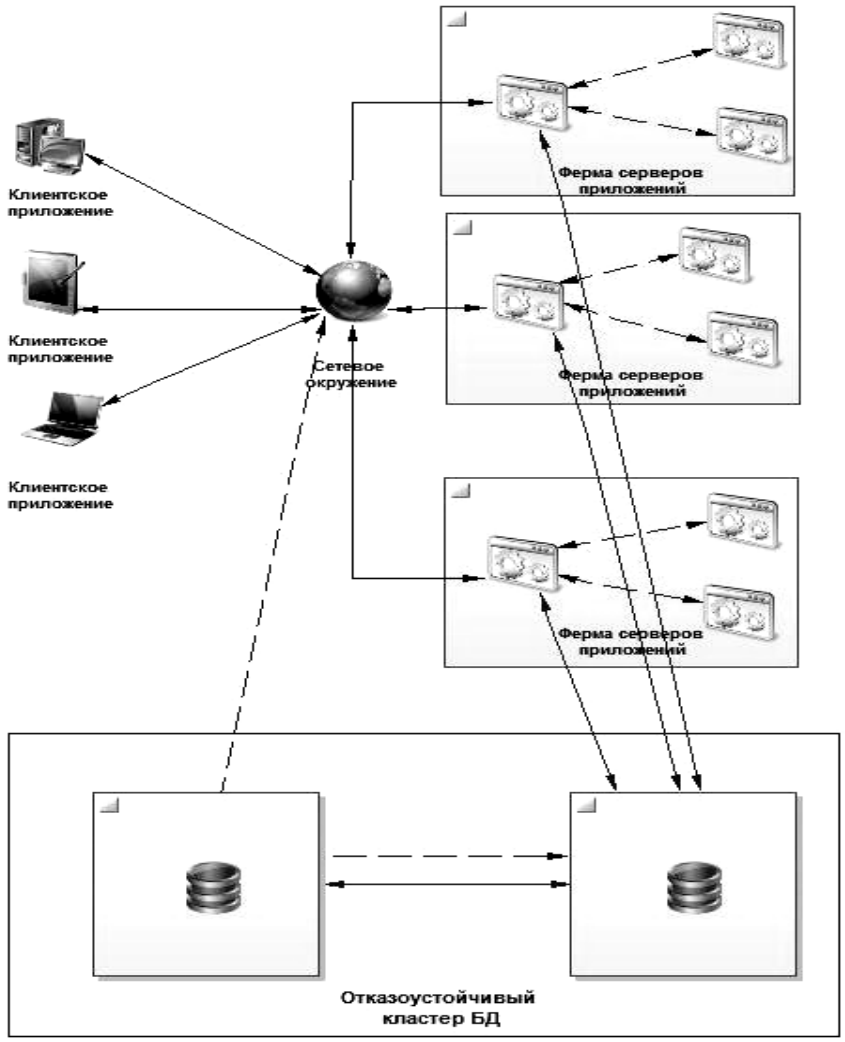

Рис. 1. Топология отказоустойчивой информационной банковской системы

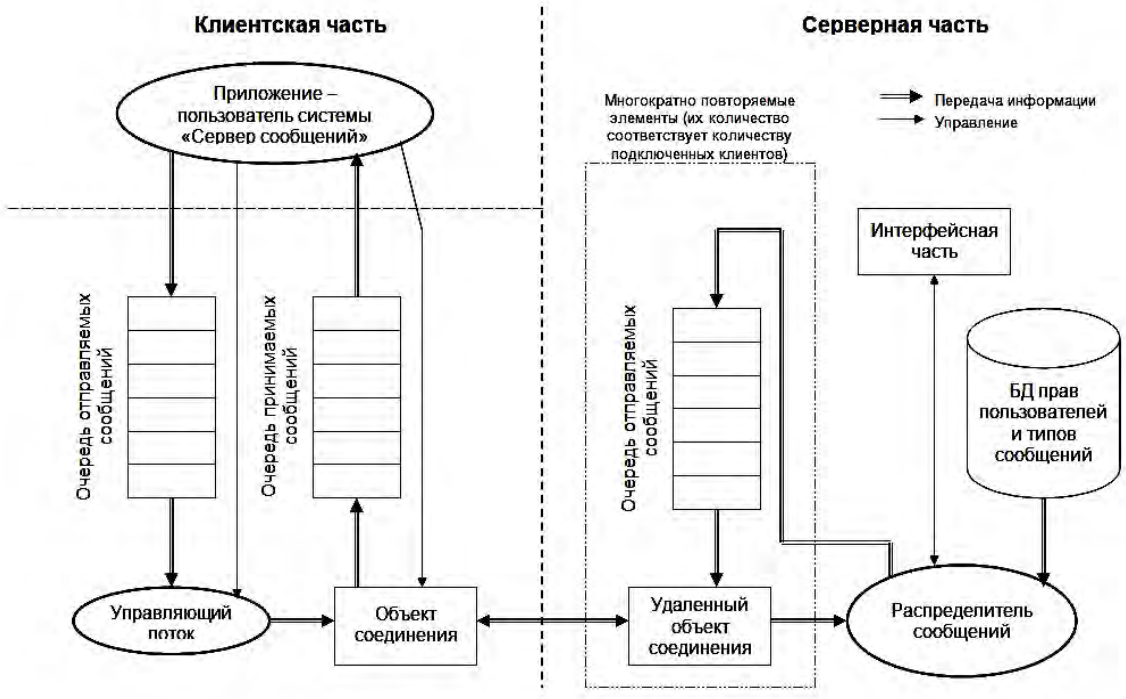

Рис. 2. Структурная схема сервера приложений

Удаленный объект соединения серверной части предназначен для обработки клиентских соединений. Он функционирует по принципу конечного автомата.

Удаленный объект соединения имеет четыре основных состояния: инициализация, аутентификация клиентского соединения, прием сообщений, деинициализация.

Через методы распределителя сообщений данный объект обеспечивает авторизацию клиентов, управляет каналом связи.

Запросы обслуживаются по методу FIFO. Связь между клиентом и сервером устанавливается через механизм аутентификации/авторизации. Логин и пароль через объект соединения передается на сервер, где происходит его верификация.

После успешной аутентификации происходит определение доступа к операциям на основании принадлежности вызывающей стороны к той или иной роли. Связь между сервером приложений и базой данных обеспечивается отдельным соединением.

Таким образом, аутентификация клиента происходит через посредника – сервер приложений, а сервер приложений использует собственные учетные данные для доступа к ресурсам, а не учетные данные клиента.

Учетные данные клиента используются для определения его принадлежности к той или иной роли.

В случае обрыва соединения клиент пытается снова установить соединение с текущим сервером приложения, по истечении времени таймаута – с резервным, имеющим наибольший приоритет. При этом все сообщения, стоящие в очереди, сохраняются и передаются после подключения к новому серверу.

В случае невозможности подключиться к серверу приложений, система адаптирует запросы для непосредственной работы с базой данных.

Коммуникации между клиентом и сервером защищаются с помощью протокола TLS. TLS использует цифровые сертификаты и криптографию с открытым ключом. Он обеспечивает шифрование данных, передаваемых по сети между клиентом и сервером приложений, а также между сервером приложением и базой данных, выявление возможных манипуляций, направленных на преодоление защиты или вмешательство в обмен, а также аутентификацию на основе сертификатов. Клиент запрашивает сервисы путем инициирования запросов к серверу. Запрос определяет действия, происходящие в конкретный момент времени. Форма запроса определяется протоколом обмена. Выполнение запроса вызывает выполнение соответствующего действия на сервере. После завершения запроса клиенту возвращается результат запроса.

В системе циркулируют сообщения трех типов: системные, предопределенные (службы сервера) и пользовательские. Системные сообщения используются для служебных целей и уведомлений об ошибках. Данные сообщения не могут быть изменены извне. Предопределенные сообщения используются для взаимодействия клиента и сервера.

Примерами подобных сообщений является информация о подключениях, изменениях прав доступа к сообщениям. Данные сообщения нельзя создать извне, но можно подписаться на получение. Пользовательские сообщения можно создавать извне.

Сообщение имеет следующий формат:

<Код сообщения> <Идентификатор пользователя> <Параметр> <Размер> <Код адресата> <Сообщение>

Описание полей сообщения приведено в табл. 1, а в табл. 2 приведены основные функции, посредством которых обеспечивается работа клиента с сервером приложений.

Таблица 1

Поля сообщения

Тип поля	**Размер, байт**	**Описание**
<Код сообщения>	2	Код сообщения
<Идентификатор пользователя>	2	Уникальный идентификатор пользователя
<Параметр>	2	Параметры сообщения
<Размер>	2	Размер сообщения
<Код адресата>	2	Код адресата или широковещательное сообщение
<Сообщение>	10+size(Сообщение)	Текст сообщения

Таблица 2

Функции работы с сервером приложений

Функция	Аргументы	Возвращаемое значение
Соединение с сервером приложений	*name* - имя пользователя для соединения. *pass* - пароль пользователя для соединения. *sadr* - адрес удаленного сервера сообщений. *port* - адрес удаленного порта. *win* - дескриптор окна (если NULL – сообщения не принимаются, а только отправляются). *buf* - адрес буфера приема окна.	RES_OK – успешное завершение. ALREADY_CONNECTED - объект соединения уже создан. MEMORY_ALLOCATION_ERRO R - невозможно выделить память под объект.
Отключение от сервера приложений		RES_OK - успешное завершение. CONNECTION_ABSENT - объект соединения уже (или еще) не существует.
Уведомление о готовности принять очередной запрос		RES_OK - успешное завершение. CONNECTION_ABSENT - объект соединения уже (или еще) не существует.
Отправить сообщение всем пользователям	type – тип сообщения text – текст сообщения size – размер сообщения	RES_OK - успешное завершение. NON_USER_MESSAGE - данный тип сообщений не может быть отправлен.
Отправить сообщение по адресу	type – тип сообщения text – текст сообщения size – размер сообщения address – адрес пользователя	RES_OK - успешное завершение. NON_USER_MESSAGE - данный тип сообщений не может быть отправлен.
Получить информацию об ошибке	text – буфер, в который будет помещен ответ	RES_OK - завершение. CONNECTION_ABSENT -объект соединения уже (или еще) не существует.
Установить время задержки при подключении	delay – время задержки	RES_OK – успешное завершение. RES_NO – ошибка выполнения.
Временный запрет приема сообщений определенного типа	type – тип сообщения, прием которых нужно приостановить	RES_OK – успешное завершение. ALREADY_FILTRED - фильтр уже установлен. VERY_MANY_FILTERS - слишком большое число фильтруемых сообщений.
Убрать запрет приема сообщений определенного типа	type – тип сообщения	RES_OK - успешное завершение FILTER_ABSENT - фильтр не установлен.
Снять все ограничения на прием сообщений		RES_OK - успешное завершение.

Выводы

Современные тенденции говорят о возрастающем внимании к операционным рискам со стороны НБУ и коммерческих банков в Украине. Стремление национальной банковской системы придерживаться рекомендаций Базельского комитета вынуждает банки выстраивать систему риск-менеджмента, в которой операционный риск занимает значительное место, так как высокотехнологичная информацион-

ная инфраструктура, стабильность и надежность работы банка, четко отлаженные бизнес-процессы и грамотный персонал являются основными конкурентными преимуществами и основой для повышения качества банковского сервиса. Предложенный механизм динамически настраиваемой инфраструктуры информационной системы существенно повышает отказоустойчивость критических узлов информационной инфраструктуры, сводя к минимуму время восстановления, что позволяет сделать систему эластичной к отказам и обеспечивает непрерывность автоматизированных бизнес-процессов финансового учреждения.

Описанная динамически настраиваемая инфраструктура реализована под руководством автора в нескольких коммерческих проектах, эксплуатируемых в финансовых и телекоммуникационных компаниях. Данное решение может быть использовано для минимизации влияния функциональной составляющей операционного риска, например в системе мониторинга наиболее значимых показателей работы банка.

Список литературы

1. Determinants of Operational Risk Reporting in the Banking Industry [Электронный ресурс]. – Режим доступа к ресурсу: http://papers.ssrn.com/sol3/papers.cfm? abstract_id=425720.

2. Иконников А.Н. Некоторые вопросы минимизации операционных рисков в АБС / А.Н. Иконников // Расчеты и операционная работа в коммерческом банке. – 2007. – № 3.

3. Готовчиков И. Системы управления банковскими операционными рисками по Basel II. Предложения по технологиям построения / И. Готовчиков // Банковские технологии. – 2007. – №7. – С. 40-44.

4. Филатов Б.Г. Определение операционного риска в банковской деятельности. Принципы управления [Электронный ресурс] / Б.Г. Филатов. – Режим доступа к ресурсу: http://www.nbuv.gov.ua/portal/soc_gum/pprbsu/texts/2009_27/09_27_11.pdf.

5. Fowler M. Patterns of Enterprise Application Architecture / M. Fowler. – Addison-Wesley Professional, 2003. – 533 p.

6. The New Basel Capital Accord. Basel Committee on Banking Supervision. April 2003.

7. Положення про організацію операційної діяльності в банках України [Электронный ресурс]. – Режим доступа к ресурсу: http://search.ligazakon.ua/l_doc2.nsf/link1/REG7880.html.

8. Методичні вказівки з інспектування банків "Система оцінки ризиків" [Электронный ресурс]. – Режим доступа к ресурсу: http://zakon2.rada.gov.ua/laws/show/v0104500-04.

9. Bernshtein Philip A. Middleware – A model for Distributed System Services / Philip A. Bernshtein // Communications of the ACM. – 1996. – Vol. 39. – № 2.

10. Гнеденко Б.В. Математические методы в теории надежности / Б.В. Гнеденко, Ю.К. Беляев, А.Д. Соловьев. – М.: Наука, 1965. – 524 с.

11. Абдуллаева Ф.Д. Метод вычисления вероятности отказоустойчивости и надежности информационной системы "Население и миграция" / Ф.Д. Абдуллаева // Вопросы защиты информации. – 2008. – №2. – С. 49-52.

12. Постанова № 474 від 28.10.2010 про набрання чинності стандартами з управління інформаційною безпекою в банківській системі України [Электронный ресурс] – Режим доступа к ресурсу: http:// zakon1.rada.gov.ua/laws/show/v0474500-10.

Поступила в редколлегию 31.08.2012

Рецензент: д-р техн. наук, проф. И.В. Шостак, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков.

ДИНАМІЧНО НАЛАШТОВУЄМА ІНФРАСТРУКТУРА ІНФОРМАЦІЙНОЇ СИСТЕМИ БАНКУ

С.Є. Голиков

У статті розглянутий механізм мінімізації інформаційно-технологічної складової операційних ризиків, що виникає внаслідок використання систем автоматизації в банківських установах України. Показано, що операційні ризики є основною причиною втрат значних грошових сум фінансових організацій. Розглянуті інфраструктурні рішення сучасних систем автоматизації банківської діяльності, виявлені їх вузькі місця. Запропоновано механізм динамічно налаштовуємої інфраструктури інформаційної системи, який дозволяє істотно підвищити її відмовостійкість та доступність. Показана можливість використання розробленого механізму для зниження функціональної складової операційного ризику. Розглянуті рішення знайшли застосування в конкретних проектах.

***Ключові слова**: операційний ризик, система автоматизації банку, відмовостійкість, резервування, доступність, сервер прикладних програм, трирівнева архітектура.*

DYNAMICALYY ADJUSTABLE INFRASTRUCTURE OF THE INFORMATION SYSTEM OF THE BANK

S.E. Golikov

The mechanism to minimize the information technology component of operational risk arising from the use of automation in the banking institutions of Ukraine is reviewed. It is shown that operational risks are a major cause of loss of significant amounts of money financial institutions. It is considered the infrastructure solutions modern systems of banking automation, revealed their bottlenecks. It is proposed a mechanism dynamically adjustable the information system infrastructure, which allows to increase its availability and accessibility. It is shown the possibility of using a mechanism designed to reduce the functional component of operational risk. Considered solutions have found applying in concrete projects.

***Keywords**: operational risk, bank automation system, fault tolerance, redundancy, availability, application server, three-tier architecture.*