Российский государственный гуманитарный университет
Russian State University for the Humanities

RGGU BULLETIN
№ 13 (75) / 11

Scientific journal

Information science. Information security.
Mathematics Series

Moscow 2011

ВЕСТНИК РГГУ

№ 13 (75) / 11

Научный журнал

Серия «Информатика. Защита информации.
Математика»

Москва 2011

УДК 94(560)
ББК 63.3(5)я54

ISSN 1998-6769

СОДЕРЖАНИЕ

Технологии

CONTENTS

Technologies

От редакции

Предлагаем Вашему вниманию очередное издание серии «Информатика. Защита информации. Математика» журнала «Вестник РГГУ». Данный выпуск посвящен скрытым каналам, исследование которых является в настоящее время актуальной задачей.

Попытки скрыть сам факт передачи информации имеют длинную историю. Способы сокрытия самого факта передачи информации получили название «стеганография». Исторически для стеганографии применялись «невидимые» чернила, точечные фотовставки и т. д. Данное направление получило вторую жизнь в наше время в связи с широким использованием сетей передачи данных. Чтобы выделить методы стеганографии, связанные с электронным представлением данных, появился термин «компьютерная стеганография». Однако в работе Шнайера стеганографические способы передачи по каналам связи получили название потайных каналов (subliminal channels). Наряду с этим появился термин «скрытый канал» (covert channel). Впервые понятие скрытого канала было введено в работе Лэмпсона в 1973 г. Канал называется скрытым, если он не проектировался, не предполагался для передачи информации в электронной системе обработки данных. Таким образом, термин «скрытые каналы» больше относится к внутрикомпьютерным телекоммуникациям. В настоящее время актуальна проблема анализа скрытых каналов всюду, где возникают ограничения на информационные потоки.

Кроме того, не следует забывать об актуальности таких вопросов, как сервис-ориентированная архитектура (СОА), анализ безопасности крупных банковских информационных систем, анализ угроз информационной безопасности и др.

Приглашаем авторов – преподавателей РГГУ и его филиалов, сотрудников научных центров, представителей большого и малого бизнеса, аспирантов, докторантов – принять участие в публикации

результатов научных исследований по современной проблематике информационных технологий и математики.

Материалы для журнала просим оформлять в соответствии с принятыми нормами, установленными для ВАКовского издания, и направлять их электронной почтой по адресу: vestnik@rggu.ru на имя ответственного редактора серии А.А. Грушо.

Д.А. Ларин

ЭТАПЫ КРИПТОГРАФИЧЕСКОЙ ДЕЯТЕЛЬНОСТИ В РОССИИ

Данная работа является первым шагом автора в данном направлении, периодизация проведена исходя из использования тех или иных криптографических идей, под влиянием которых находилась отечественная криптография в определенные временные этапы. В некоторой степени учтены организационные изменения, происходившие в криптографических службах нашей страны. Безусловно, необходимо продолжать работу в данном направлении. Итак, в криптографической деятельности России предлагается выделить следующие этапы.

Ключевые слова: тайнопись, криптография, шифр, стеганография.

I. IX в. – 1549 г.

Подробнее о становлении системы связи и защиты информации в древнерусском государстве рассказано в статье автора[1].

II. 1549 г. – конец XVII в.

Создание первой в нашем отечестве регулярной шифровальной службы в Посольском приказе. Активное использование шифров для защиты русской дипломатической переписки. Учреждение Ямского приказа с целью усовершенствования системы связи организация международных линий почтовой связи. Первый опыт организации перехвата иностранных сообщений.

Активная внутри- и внешнеполитическая деятельность Ивана IV Грозного и связанные с ней войны оказали значительное влияние на становление и развитие шифровального дела. К России

присоединяются Казанское и Астраханское ханства, а впоследствии значительные территории в Сибири, устанавливаются новые политические, экономические и торговые связи с Европой и странами Востока. В 1549 г. учреждается Посольский приказ – первое внешнеполитическое учреждение в нашем отечестве. Эта организация также занималась разведкой и контрразведкой, ведала вопросами деятельности иностранцев, в том числе купцов, живших в России татар, изготовлением карт, обменом и выкупом пленных, управлением рядом территорий на юго-востоке страны, некоторыми категориями служилых людей, а также отдельными поручениями. Именно в недрах Посольского приказа была создана первая в России регулярная шифровальная служба, на постоянной основе начинали активно использовать криптографические методы для защиты дипломатической переписки. На службе в Посольском приказе состояли и специалисты, разрабатывавшие шифры, которые тогда называли «азбуками», «цифирью», «цифрами»[2].

О применяемых в эту эпоху системах шифрования см. упомянутую выше статью. Здесь же отметим, что это была эпоха абсолютного господства шифров простой замены, иногда использовались и шифры перестановки, но весьма примитивные. Так, например, русский посол в Грузии К.П. Савин в 1597–1598 гг. применял разбивку открытого текста на слоги и перестановку букв в каждом слоге[3].

Продолжала совершенствоваться в России и система передачи информации. Для эффективного управления ямской гоньбой еще в 1516 г. была создана Ямская изба, которую в 1550 г. преобразовали в Ямской приказ – центральное почтовое учреждение России. Для эффективной передачи военной информации служило упорядочение несения службы на границах Московского государства. 16 февраля 1571 г. был утвержден Приговор (устав) «О станичной и сторожевой службе», установивший правила доставки гонцами и сторожами имеющих государственное значение вестей из столицы на места и обратно, он также содержал параграфы о передаче экстренных сообщений.

Таким образом, можно отметить, что «в XVI в. в России впервые сложилась довольно стройная система связи высших органов управления централизованным государством, обеспечивавшаяся специальными категориями служилых людей следующих структур исполнительной власти:

– Ямского приказа (почтового ведомства) – организационное обеспечение системы ямской гоньбы и доставка корреспонденции второстепенного значения;
– Разрядного приказа (военного ведомства) – доставка особо важной правительственной корреспонденции военными курьерами (гонцами);

– Посольского приказа (внешнеполитического ведомства) – организация и обеспечение криптографической защиты внешне- и внутригосударственной переписки»[4].

Начало XVII в. ознаменовалось Смутой, в результате система государственного управления страной была практически полностью разрушена. С приходом к власти в 1613 г. первого царя династии Романовых – Михаила Федоровича началось возрождение разрушенного хозяйства. Но при больном и бездеятельном сыне огромная власть фактически находилась в руках его отца – патриарха Филарета. Он лично занимался делами Посольского приказа и даже разработал несколько дипломатических шифров. В 1633 г. патриарх Филарет написал «для своих государевых и посольских тайных дел» особую азбуку и «склад затейным письмом»[5].

В 1634 г. между Россией и Польшей был заключен первый в истории нашей страны договор о международной перевозке корреспонденции, согласно которому гонец мог иметь при себе до шести провожатых. В 1665 г. заработала почтовая линия между Москвой и Ригой. Чуть позже было заключено почтовое соглашение со Швецией[6].

При Алексее Михайловиче шифрование получает еще более широкое распространение. Сам царь использовал в личной переписке разнообразные шифры. Как правило, это была простая замена из специально вымышленных знаков. С 1654 по 1676 г. криптографическая деятельность в России сосредоточилась в Приказе тайных дел. Этот приказ «входил в область дипломатическую, военную, полицейскую, финансовую и отправлял множество еще других функций, не поддающихся никакой классификации»[7]. Здесь проводились организационные мероприятия по совершенствованию работы шифровальной службы с целью недопущения утечек информации. Так, подьячие Приказа тайных дел и посольские дьяки, поддерживавшие связи с царскими представителями за границей, активно прибегали к зашифрованной переписке («затейное письмо»). Ключ для расшифрования этих посланий не записывался, его заучивали наизусть. Существовали различные варианты секретного письма, и, как положено по правилам конспирации, никто из подьячих не должен был знать всех вариантов тайнописи.

В это же время была организована система регулярного перехвата и перлюстрации (тайное вскрытие и копирование) корреспонденции зарубежных представителей, находившихся в России. В.О. Ключевский пишет об этом, ссылаясь на свидетельства иностранцев: «письма... якобы вскрывались, прочитывались и потом уничтожались»[8]. Здесь стоит высказать гипотезу, что уничтожали как раз шифрованные письма, которые в Приказе тайных дел прочитать не могли, и действовали по принципу «так не доставайся ж ты никому».

III. Петровская эпоха (конец XVII в. – 1725 г.)

Появление шифров разнозначной и пропорциональной замены, простейших номенклаторов, первое упоминание о криптоанализе, реформа криптографической службы. Активное использование криптографических методов защиты информации для защиты не только дипломатической, но и военной и внутриполитической информации. Разработка нормативных документов, регламентирующих организацию конфиденциальной связи, работу с шифрами и их хранение, а также подбор кадров для шифровальной службы.

Император Петр I (1672–1725) вошел в историю России как великий реформатор. Главным итогом петровских реформ стало преодоление серьезного отставания России от европейских держав в военной, экономической и политической областях. Разумеется, эффективное государственное управление, военные победы и дипломатические успехи были бы невозможны без активной криптографической деятельности. Петр это отлично осознавал и уделял большое внимание криптографии как надежному средству сохранения государственных секретов.

В Петровскую эпоху в России велась активная криптографическая деятельность. Шифрование стало основным видом защиты информации, хотя продолжали использоваться и другие методы: стеганография, физическая защита, условная сигнализация и т. п. При этом если в предыдущие времена практически вся шифрпереписка была посвящена дипломатическим вопросам, то при Петре I стала активно шифроваться также военная и внутриполитическая информация. Шифрованная связь становится основным средством управления центральным государственным аппаратом и местными органами власти, армией и флотом.

Для организации связи были созданы ряд учреждений (Кабинет его императорского величества, Посольская канцелярия, Коллегия иностранных дел и т. д.) со строгим распределением задач между ними. Деятельность данных учреждений была регламентирована законодательно. Широко практиковались организационно-административные методы защиты информации, такие как особый подбор кадров для осуществления криптографической деятельности и строгий контроль за их работой, организация пропускного режима в помещения, где производилось шифрование и расшифрование или хранились шифры, обеспечение охраны гонцов и курьеров и т. д. Из новшеств в области связи отметим, что для обеспечения управления флотом и прибрежными районами страны в качестве средства связи стали широко использоваться суда.

К концу XVII в. Россия становится самой крупной европейской державой, занимающей огромную территорию от Днепра на западе и до Тихого океана на востоке. Для организации эффективного управления этой огромной территорией в первую очередь была необходима быстрая, надежная и конфиденциальная связь. Основным средством передачи сообщений в это время была почта. Наиболее распространенным методом защиты информации являлась физическая защита. Ямщики отвечали за сохранность почты и целостность печатей.

С созданием регулярной армии возникла настоятельная потребность в необходимости совершенствования управления войсками в мирное и военное время. Уже во время азовских походов против турок в 1695 и 1696 гг. Петр I впервые организовал работу военно-полевой почты, возглавлял ее первый русский почтмейстер А.А. Виниус. Отправления по линии этой почты назывались чрезвычайными[9].

Особое значение для защиты информации во время военных походов и дипломатической деятельности в Петровскую эпоху приобрела криптография. Шифры использовались во время Азовских походов и Великого посольства. Однако наибольшие потребности в осуществлении криптографической деятельности возникли в начале XVIII в., когда за рубежом появились дипломатические представительства России, с которыми необходимо было поддерживать связь. В 1700 г. началась Северная война со Швецией, что потребовало организации эффективного управления войсками на весьма значительном театре военных действий, а также координации действий с союзниками, и здесь нельзя было обойтись без шифров. И наконец, реформы системы государственного управления огромной страной также требовали обеспечения секретности переписки внутри страны.

Теперь рассмотрим шифрсистемы, которые применялись в России в эпоху Петра Великого. Как и в прежние годы, основным шифром на Руси была простая замена, т. е. знаки открытого текста заменялись буквами (при этом буквы могли принадлежать алфавиту как открытого текста, так и другой азбуки), цифры или специально придуманные знаки. При этом следует отметить, что в шифрах в то время употреблялись только привычные нам арабские цифры, так как в начале XVIII в. Петром была выведена из употребления архаичная буквенная кириллическая нумерация, заимствованная у греков. В качестве знаков шифрованного текста употреблялись и буквенные сочетания.

Тексты, которые надо было зашифровать, могли быть написаны на русском, французском, немецком, а иногда даже на греческом

языках. Как известно, Петр I прекрасно владел несколькими европейскими языками, в то же время на государственной службе состояло много иностранцев и именно в переписке с ними в основном использовались немецкий и французский. При этом следует отметить, что с точки зрения стойкости предпочтительнее использование русского языка. За границей было очень мало людей, владевших русским языком, а знание лингвистических особенностей языка может существенно помочь криптоаналитику в дешифровании.

Новым явлением для российских шифров Петровской эпохи по сравнению с предыдущими временами стало наличие во многих из них «пустышек» – знаков шифрованного текста, которым не соответствует никакой знак открытого текста, т. е. они не несли никакого смысла. Как правило, пустышек было немного, обычно 5–8 знаков. Наличие «пустышек» увеличивает стойкость шифра, так как они дают криптоаналитику неверную информацию о количестве знаков в алфавите открытого текста, разбивают структурные лингвистические связи открытого текста и изменяют статистические закономерности, т. е. именно те свойства текста, которые используют при дешифровании шифра простой замены. Кроме того, «пустышки» увеличивают длину шифрованного текста по сравнению с открытым, что усложняет их взаимное сопоставление. Кроме того, в некоторых случаях отдельные знаки применялись для зашифрования точек и запятых, содержавшихся в открытом тексте, также для этого могли использоваться «пустышки». Это особо оговаривалось в кратких правилах пользования шифром.

Вскоре к обычному алфавиту простой замены стали добавлять обозначения для наиболее употребительных слогов, слов и целых фраз, т. е. стали употребляться номенклаторы. Петровские номенклаторы были весьма простыми, они содержали небольшой словарь, называвшийся «суплемент» и содержавший некоторое количество слов (имен собственных, географических наименований или каких-то устойчивых словосочетаний, которые могли часто использоваться в открытых текстах корреспондентов, использовавших данный шифр).

Шифр Петровской эпохи представлял собой лист бумаги, на котором от руки были написаны ключ – таблица замены (обычно под горизонтально расположенными в алфавитной последовательности буквами кириллицы или иного алфавита, подписаны соответствующие элементы шифроалфавита), а также суплемент (если это был номенклатор). Ниже могли помещаться пустышки и краткие правила использования шифра. Еще одной особенностью петровских шифров является то, что шифроалфавит мог составляться из символов разных типов, например смеси букв разных алфавитов, цифр и т. д.

Главным новшеством в шифровальном деле, появившемся при Петре I, стало применение более сложных, чем простая замена, шифров. Так в 20-х годах XVIII в. от шифров простой замены переходят к замене пропорциональной, когда наиболее часто встречающимся знакам открытого текста присваиваются несколько шифробозначений, что затрудняет использование «классического способа» криптоанализа шифра простой замены – частотного анализа. Еще раньше, примерно с 1708 г., в России начали применяться шифры разнозначной замены (когда для замены одного знака открытого текста используются один или два знака шифротекста). Следует отметить, что шифры такого типа являются несколько более стойкими, чем «классическая» простая замена, но они чувствительны к ошибкам при шифровании (как к замене нужной буквы на другую, так и к пропуску или вставке лишней буквы).

В Петровскую эпоху началась нормативно-правовая регламентация криптографической деятельности. Так, в 1716 г. был принят «Устав воинский», первый в истории России документ подобного рода. В соответствии с Уставом учреждены должности «адъютантов, ординарцев, курьеров для передачи и доставки секретных донесений»[10], а также отредактированы «Правила действия военно-полевой почты». Военная полевая почта стала активно использоваться для быстрого обмена конфиденциальной корреспонденцией между крупными воинскими соединениями и вновь создаваемыми центральными органами управления армией и флотом – Военной коллегией и Адмиралтейств-коллегией. Для доставки особо важной и срочной воинской корреспонденции в адрес императора или президентов оборонных коллегий при главнокомандующих армиями были учреждены должности военно-полевых курьеров. Следует отметить, что «собственноручно отредактированные Петром I “Правила действия военно-полевой почты” являются, по сути, первым дошедшим до нас документом той эпохи, в котором было четко и определенно сформулировано назначение военно-курьерской связи»[11].

Преобразования системы государственного управления страной, осуществляемые Петром I, касались и повышения эффективности системы связи. В 1720 г. был введен Генеральный регламент, определяющий процесс функционирования системы управления России, в котором впервые четко устанавливался порядок работы с правительственной корреспонденцией. Вот как это выглядело:

«В соответствии с регламентом, вся правительственная корреспонденция в зависимости от степени ее важности и срочности была разделена на определенные группы. Так, особо важные правительственные документы, например именные указы императора с его личными распоряжениями, всеподданнейшие донесения и доклады

на его имя, реляции, манифесты направлялись по каналам "необыкновенной", "чрезвычайной" почты, через правительственных гонцов и нарочных курьеров. Особо важную корреспонденцию из личной канцелярии императора доставляли кабинет-курьеры, входившие в штат Кабинета его императорского величества и имевшие воинские звания обер-офицерского состава. Когда же небольшая группа кабинет-курьеров (по штату 6 человек) не справлялась с доставкой всей исходящей из Кабинета особо важной корреспонденции, для выполнения обязанностей правительственных курьеров широко привлекались военные чины от капрала до полковника, в зависимости от важности даваемого им поручения. А по делам особо важным, как, например, доставка писем императора главам иностранных государств, привлекались генералы (в петровское время преимущественно П.И. Ягужинский и Л.К. Нарышкин). Сенат также имел в своем распоряжении прикомандированных армейских офицеров, которых использовал в качестве курьеров для доставки адресатам своей исходящей корреспонденции»[12].

Именно в Петровскую эпоху свои первые шаги сделал российский криптоанализ[13], стали развиваться методы тайного перехвата информации, осуществлялась агентурная добыча криптографических секретов противника. При этом важно отметить, что занимался оценкой стойкости российских шифров не кто-нибудь, а сам царь! Приведем две цитаты в подтверждение этому. Петр I писал фельдмаршалу Огильви по поводу одного шифра (цифири, как их тогда называли): «А которую вы перво прислали, и та не годна, понеже так, как простое письмо, честь можно»[14]. Вот еще одна нелестная оценка царя одного из российских шифров: «Сия цифирь зело к разобранию легка»[15]. Стали проявлять в России интерес и к шифрам иностранных государств, хотя, конечно, о регулярном дешифровании иностранной переписки речь пока не шла, однако заинтересованность в получении информации таким методом уже была. Русским дипломатам, разведчикам и другим представителям за границей предписывалось добывать любую информацию, касающуюся шифров, организации связи, открытых текстов (против атаки «открытый текст – шифрованный текст» подавляющее большинство шифров того времени было не устойчиво). На этих лиц и их зарубежную агентуру также возлагалась задача организации перехвата иностранных сообщений за пределами России.

В это время пришло понимание важности такого источника информации, как шифрованная переписка иностранных государств. Полученная информация могла быть весьма полезна российскому руководству, дипломатам, военным. Велся перехват сообщений и внутри России, совершенствовались методы перлюстрации, осу-

ществлялась цензура, вводились ограничения на почтовые сообщения, и здесь российские власти столкнулись с шифрами (дело царевича Алексея). Таким образом, можно отметить появление в России негосударственной криптографии. Заметим, что Петр I считал шифры монополией царя российского. Он строго наказывал своих подданных за использование «негосударственных» шифров (цифирей). Однако частные лица все же пользовались собственными шифрами. Среди них можно отметить царевну Софью Алексеевну, которая использовала шифр в переписке со своим фаворитом князем В.В. Голицыным, а также царевича Алексея и его сподвижников.

В заключение скажем, что Петр Великий был первым из российских правителей, кто предельно ясно осознавал важность криптографической деятельности для обеспечения безопасности государства. Во время его правления впервые в истории отечественной криптографии к осуществлению криптографической деятельности (в том числе к составлению шифров) привлекалось все высшее руководство России, включая самого царя.

IV. 1725 г. – 30-е годы XIX в.

Развитие систем многоалфавитной замены. Широкое применение кодов и сложных номенклаторов. Введение «генеральной цифири». Появление «черного кабинета» – дешифровальной службы. Активное чтение иностранной шифрпереписки. Вскрытие шифров Наполеона и его генералов во время Отечественной войны 1812 г.

После смерти Петра I в течение XVIII в. Происходила, пусть даже и не всегда удачно, дальнейшая модернизация России. Передовые идеи часто наталкивались на сопротивление подавляющей части дворянства и не находили реализации в исторических условиях того времени. Но в области криптографической деятельности Россия неуклонно двигалась вперед.

В годы царствования императриц Екатерины I, Анны Иоановны, Елизаветы Петровны и тем более Екатерины II сеть шифрованной документальной связи неоднократно расширялась. В отличие от индивидуальных шифров, с середины XVIII в. стал использоваться меняющийся примерно раз в два года главный шифр (так называемая генеральная цифирь). Имелись также отдельные шифры у различных корреспондентов для связи между собой. Для ведения переписки, в том числе и шифрованной, все крупные гражданские и военные деятели России имели специальный штат канцелярских работников. Шифрование и расшифрование текстов, сообщений,

депеш производились секретарями-переводчиками, владеющими двумя-тремя иностранными языками.

Вся посольская и внутренняя конфиденциальная переписка шифровалась и держалась в строгом секрете. Большое внимание соблюдению тайны уделялось и в самой Коллегии иностранных дел в отношении лиц, работавших по шифровальной части. С 1744 по 1791 г. четырежды документально подтверждались требования к этим лицам. В основном они касались:

- неразглашения военной тайны;
- строгого режима допуска в апартаменты, занятые секретной экспедицией;
- ответственности секретарей экспедиции за деятельность переводчиков и их личные контакты;
- запрещения общения нижних чинов экспедиции с иностранцами и т. д.

В Коллегии иностранных дел велся тщательный учет всех цифирей (шифров). В случае даже подозрения на компрометацию генеральной цифири немедленно издавался императорский указ о выводе данного шифра из действия и замене его другим[16]. В 30-е годы XVIII в. в России появляются совершенно новые системы шифрования: алфавитные, позже – неалфавитные коды. В алфавитном коде текст и шифробозначения нумеруются параллельно друг другу. Это-то и было слабым местом. Данная система облегчала дешифрование, повысить стойкость можно было путем перемешивания шифробозначений, т. е. посредством перехода к неалфавитному коду. Также продолжали усложняться путем увеличения кодового словаря номенклатуры. Подобные шифры стали основными в рассматриваемый период времени, активно использовались и шифры многоалфавитной замены.

Система связи в Российской империи продолжает совершенствоваться. В 1781 г. управление всей внутригосударственной почтой России сосредоточилось в одном ведомстве – Санкт-Петербургском почтамте, или почтовом департаменте, подчинявшемся Коллегии иностранных дел, а в 1802 г. причисленном к Министерству внутренних дел. Почтовые тракты (к концу XVIII в. их общая протяженность составляла 33 тыс. верст) являлись основными магистралями связи, по которым правительственная корреспонденция перевозилась курьерами, а ведомственная и частная – почтальонами. Для повышения эффективности доставки правительственной, дипломатической и военной корреспонденции 17 декабря 1796 г. указом императора Павла I был создан Фельдъегерский корпус. Корпус стал специальной воинской частью, предназначенной для несения службы связи и выполнения особых поручений императора[17].

Дешифровальная служба России достигла больших успехов во время правления императриц Елизаветы Петровны и Екатерины II, в это время читалась практически вся шифропереписка иностранных послов при российском дворе. 40-е годы XVIII столетия, а точнее 1742 г., можно с полным правом считать временем создания дешифровальной службы России. К этому периоду сложилась определенная система дешифровальной деятельности: была создана служба перехвата и перлюстрации секретной шифропереписки иностранных корреспондентов, организованы ее дешифрование, перевод, доклад сообщений в высшие инстанции. Была осознана необходимость организации криптографической службы как единой слаженной системы, придания ей научной базы. Научный подход и активное заинтересованное внимание руководителей государства к специальной службе позволили России добиться быстрых и важных успехов в дешифровании корреспонденции Франции, Англии, Германии и других стран.

Дешифровальную службу России при Елизавете возглавлял один из первых академиков Санкт-Петербургской академии наук, математик, работавший в области математического анализа и теории чисел, Христиан Гольдбах (1690–1764). В 1725 г. он приехал в Россию и был назначен профессором математики в Санкт-Петербургской академии наук, в 1726–1740 гг. исполнял обязанности конференц-секретаря академии. В 1742 г. перешел на службу в Коллегию иностранных дел и переехал в Москву. Особенно он прославился при раскрытии шифров посла Франции в России Д. Шетарди. В результате тяжелой и кропотливой работы Гольдбах приобрел огромный опыт в искусстве дешифрования, что позволяло ему раскрывать чужую «цифирь» в короткие сроки. Курировал же работу российского «черного кабинета» (служба перлюстрации и дешифрования) лично канцлер А.П. Бестужев-Рюмин. Дешифровальной работой занимался Франц Эпинус (1724–1802), живший в России с 1757 г., известный математик и физик, изучавший математическими методами электромагнитные явления, также большой вклад в отечественную криптографическую науку внесли Е. и Ф. Каржавины, занимавшиеся как криптоанализом, так и составлением шифров.

Необходимость скрыть от чужих глаз тексты секретных сообщений государственных деятелей, послов, военачальников, разнообразных агентов вела, разумеется, к повышению стойкости шифров. В связи с этим в середине 50-х годов XVIII в. «цифирные азбуки» увеличиваются в объеме, включая в себя более тысячи величин (так личный код Николая II в начале XX в. насчитывал 10 тыс. величин). Кодовые словари шифров включают буквы, слоги, географические

названия, имена, даты. Шифровальщики отказываются от знаков и целиком и полностью переходят к цифрам; создаются особые знаки, так называемые скрытые пустышки (при их дешифровке отдельные элементы текста могут ничего не значить).

История криптографической при Екатерине II была тесно связана с графом Н.И. Паниным. В 60–80-е годы XVIII в. по его указаниям вводились в обиход разнообразные новые шифры. Не меньше внимания граф уделял дешифрованию переписки иностранных посланников в России со своими королями. Н.И. Панин весьма преуспел в организации перлюстрации и дешифрования бумаг послов и других иностранных представителей. Как и при Елизавете, наши дешифровальщики читали практически всю входящую и исходящую иностранную шифропереписку. Важную роль сыграли они в достижении победы над Наполеоном[18].

V. 30-е годы XIX в. – конец 1920-х годов

Появление новых средств связи, профессии шифровальщика, разработка новых шифров, массовое использование шифровальных приборов, внедрение новых способов перехвата. Образование шифровальных служб в МВД и военном ведомстве. Активная криптографическая деятельность революционеров и борьба с ними правоохранительных органов, появление дешифровальной службы в МВД. Коренная реорганизация системы государственного управления (в том числе и криптографических служб) после Революции 1917 г.

В 1794 г. гениальный русский изобретатель И.П. Кулибин сконструировал семафорный (оптический) телеграф и разработал код к нему. Записанный в виде одной таблицы код упрощал работу по передаче сообщений. Это позволяло быстрее передавать нужную информацию. Оптический телеграф широко применялся в России всю первую половину XIX в. В 1808 г. офицер русского военно-морского флота А. Бутаков разработал свою систему семафорного телеграфа. Она успешно была применена в 1810 г. русскими моряками эскадры, действовавшей на Средиземном море под флагом вице-адмирала Д.Н. Сенявина. В 1824 г. между Санкт-Петербургом и Шлиссельбургом была проложена опытная линия семафорной связи (в ту пору их в России называли горизонтными) по проекту генерал-майора П.А. Козена. Линия проработала до 1836 г. Она служила для передачи сообщений о движении судов по Ладожскому озеру.

Первая правительственная линия оптического телеграфа между Санкт-Петербургом и Кронштадтом (Зимний дворец – Стрельна –

Ораниенбаум – Кронштадт) протяженностью 30 км была оборудована французским инженером Жаком Шато в 1833 г. Интересно отметить, что Шато сумел существенно упростить телеграфный код. Зимний дворец в 1835 г. получил прямую оптическую телеграфную связь с Царским Селом и Гатчиной. Тогда же международные события побудили русское правительство выделить средства для строительства линии оптического телеграфа от Санкт-Петербурга до Варшавы. В течение 1835–1838 гг. была сооружена самая длинная в мире линия семафорного телеграфа. Еще год ушел на ее испытания. Официальное открытие линии круглосуточного действия состоялось 20 декабря 1839 г. На линии длиной 1200 км было 149 промежуточных станций в виде типовых башен высотой 21,5 м с металлическим шестом высотой 3 м, через которые сигнал проходил за 15 минут. Правительственная шифрованная депеша, состоявшая из 45 знаков, передавалась из Санкт-Петербурга в Варшаву за 22 минуты. В штате линии числилось 1908 человек. В зависимости от числа промежуточных станций и погодных условий на передачу сигнала по российским линиям оптического телеграфа затрачивалось от 2 до 15 минут. По линиям длиной 1000–1200 км депеша из 45–100 знаков передавалась за 22–35 минут. Оптический телеграф просуществовал в России около полувека, примерно до середины 1850-х годов. Он сыграл значительную роль в развитии внутренних коммуникаций как средство оперативного управления исполнительными органами государства в мирное и военное время.

Первый практически пригодный электромагнитный телеграф был создан российским подданным бароном Павлом Львовичем Шиллингом фон Канштадтом19, выдающимся ученым и изобретателем. Этот аппарат он публично продемонстрировал в 1832 г. В основе действия этого аппарата находился эффект отклонения магнитной стрелки в результате воздействия электромагнитного поля от электрических проводов. В 1828 г. прообраз будущего электромагнитного телеграфа был готов и испытан. Он представлял собой двухпроводный однострелочный телеграф. Аппарат содержал все основные узлы, необходимые для телеграфирования: источник питания – вольтов столб (или столбец, как его называл сам Шиллинг); передатчик, подключавший к каждому из двух линейных проводов то один, то другой полюс батареи; двухпроводную линию; коммутатор, производящий переключение с приема на ожидание передачи, и, наконец, приемник. Для передачи латинского алфавита и цифр Шиллингом был разработан специальный код из комбинаций разного числа (от одного до пяти) последовательных сигналов, посылаемых током разного направления. Первая публичная демонстрация телеграфа Шиллинга происходила 9 (21) октября

1832 г. Передатчик был установлен на одном конце этажа, а приемник – на другом, в рабочем кабинете Шиллинга, на расстоянии немногим более 100 м. Первая телеграмма, состоящая из десяти слов, на глазах у присутствующих была принята по электромагнитному телеграфу лично П.Л. Шиллингом моментально и верно. Несмотря на большой интерес общественности к новому изобретению, правительство не торопилось с его внедрением. Только в 1836 г. в России был наконец образован под председательством морского министра «Комитет для рассмотрения электромагнетического телеграфа», предложивший Шиллингу установить телеграф в здании Главного адмиралтейства с целью длительных испытаний его в условиях, близких к эксплуатационным. Аппараты располагались в противоположных концах длинного здания, провода были проложены частично под землей, частично под водой. Но из-за неполадок линия так и не была введена в действие. В мае 1837 г. комитет предписал Шиллингу устроить телеграфное сообщение между Петергофом и Кронштадтом, для чего надо было составить проект и смету. Выполнить задачу П.Л. Шиллинг не успел, так как летом 1837 г. ученый скончался.

Большое влияние на развитие криптографии в XIX в. оказал научно-технический прогресс, в частности изобретение телеграфа, телефона и радио. Скорость передачи информации резко увеличилась – шифровать надо было быстро и без ошибок. Это привело к разработке шифровальных приборов (шифраторов) для автоматизации процессов шифрования и расшифрования. Заметим, что впервые проводной телеграф для военной связи в ходе боевых действий был применен русскими войсками во время Крымской войны 1853–1856 гг. Всю вторую половину XIX в. шла активная «телеграфизация» России, протяженность телеграфных линий неуклонно росла. Электрический телеграф стал основным средством связи в системе государственного управления Российской империей. С начала 1880-х годов в стране начинается эксплуатация телефонных аппаратов. Телефонная связь бурно развивается и подобно телеграфной охватывает все большие пространства России. В военном ведомстве появляются подразделения, в задачу которых входит создание полевых сетей телеграфной, а впоследствии и телефонной связи в районах боевых действий[20].

Сопряжение аппаратуры шифрования с техникой, передающей телеграфные сообщения, существенно повысило требования к быстродействию процесса шифрования. Шифрование при непосредственной передаче сообщения должно производиться в том темпе, который диктует телеграфный аппарат. Телеграфная связь значительно увеличила объем передаваемых сообщений (в том чис-

ле и секретных). Потребовалась разработка новых шифров с легкой сменой ключей. Это также стимулировало развитие криптографии.

В описываемый период государственные шифровальные службы ввели новые шифры (биграмный, биклавный, новые варианты многоалфавитной замены, «лямбда» и др.), продолжая активно использовать коды. Кодовые таблицы объемом до 1000–1200 словарных величин было принято называть словарными ключами и в зависимости от словаря конкретного кода – французскими, русскими, немецкими. Их применяли в Военном министерстве и МВД, в МИД и некоторых других гражданских ведомствах. Так, с помощью кодов, введенных в действие во второй половине 1860-х годов, вели секретную переписку Министерство финансов, Министерство путей сообщения, Министерство государственных имуществ, Государственный контроль, Государственная таможенная служба[21].

В конце XIX в. в России были предприняты попытки создания аппаратов для автоматического шифрования телеграфных сообщений. Так, в 1879 г. главный механик Петербургского телеграфного округа И. Деревянкин предложил оригинальный прибор по шифрованию телеграмм, который он назвал «криптограф». Это устройство напоминало известный шифратор эпохи Возрождения – диск Альберти. Прибор представлял собой два диска, один из которых был подвижным. Применялись и другие примитивные шифровальные приборы, в основном реализующие многоалфавитную замену (линейки, диски и т. п.). В качестве примеров подобных приборов можно привести механический прибор «Скала», предназначенный для облегчения работы с шифром «лямбда», и разработанное в 1916 г. подпоручиком Попазовым шифровальное устройство, впоследствии названное «Прибор Вави». Устройство по своей идее было похоже на широко известный шифратор Джефферсона[22].

Во второй половине XIX в. криптографическая служба России была вновь реорганизована. Она была создана (кроме МИД) еще в двух ведомствах – военном и внутренних дел (в департаменте полиции). Сфера использования криптографии существенно расширилась. Стали прибегать к шифрованию переписки жандармерия и гражданские ведомства. Иногда шифры употреблялись для совершенно особых миссий. Так, сенатор, тайный советник Тапильский, посланный Александром II из Петербурга в Москву к митрополиту Филарету с просьбой составить царский манифест 1861 г. об освобождении крестьян, имел личный шифр для почтовых отправлений. Деликатность миссии сенатора заключалась в том, что царь считал нежелательным разглашение факта поручения духовному лицу светского дела. Разрабатывались специальные агентурные шифры. Шифровались и несекретные документы (на так называемых клю-

чах специального назначения) с целью недопущения утечки информации к «третьим лицам» (журналистам и др.). Особые шифры разрабатывались для разведчиков и агентов.

Активно работали и российские криптоаналитики: как и в предыдущие годы, читалась практически вся шифропереписка иностранных представительств, при этом кроме аналитических активно использовались агентурные методы добычи криптографических секретов[23].

В 1895 г., благодаря русскому ученому А.С. Попову, мир получил новый способ связи – радио. Оно практически сразу же стало применяться для обмена сообщениями в русской армии и на флоте. Во время русско-японской войны 1904–1905 гг. впервые в мире начали применять радиоразведку (наблюдение за радиосетями противника, перехват и дешифрование вражеских радиограмм) и радиоэлектронную борьбу (постановка помех с целью срыва радиосвязи противника). Приоритет в использовании этих новых видов боевых действий принадлежит российскому военно-морскому флоту. Фактически русские моряки начали войну в новом измерении – радиоэфире[24]. Во второй половине XIX – начале XX в. помимо государственных организаций в России криптографическую деятельность активно осуществляли различные подпольные организации, оппозиционные власти, такие как «Народная воля», РСДРП, БУНД (еврейская подпольная организация), эсеры, анархисты и т. д. При этом революционерами были организованы сети засекреченной связи, нередко насчитывающие несколько сотен корреспондентов. Шифропереписка велась не только внутри России, но и за ее пределами. Революционеры использовали различные шифры (Виженера, Гронсфельда, «гамбетовские», книжные, перестановки, стихотворные, по слову и т. д.). Также подпольщики активно использовали стеганографию (невидимые чернила на основе природных и специально синтезированных химических компонентов, сокрытие сообщений в переплетах книг и т. п.). В ответ правительство создало организации, боровшиеся с подпольной криптографией. Деятельность этих организаций оказалась достаточно эффективной. Она была окружена большой секретностью и находилась под непосредственным контролем высших должностных лиц государства (включая императора). Успехи в ней щедро поощрялись как материально, так и морально. В 1898 г. в Особом отделе Департамента полиции была создана дешифровальная служба МВД, которая добилась значительных успехов в криптоанализе шифропереписки революционеров[25].

В результате Октябрьской революции 1917 г. страна оказалась расколота, криптографическую деятельность активно вели обе

стороны (красные и белые)[26]. После революции и окончания Гражданской войны советское правительство обратило внимание на развитие криптографии. В структуре ВЧК в 1921 г. был создан спецотдел, который стал заниматься как разработкой шифров для правительства, дипломатов и армии, так и дешифрованием сообщений иностранных государств и противников СССР (в частности, белой эмиграции). На этом поприще были достигнуты серьезные успехи, применялись как аналитические методы, так и оперативно-агентурные. В 1920 – 1930-х годах тайные операции по добыче иностранных криптографических секретов проводились советскими спецслужбами в разных регионах мира и принесли очень хорошие результаты[27]. Что касается защиты информации, то основным дипломатическим шифром в СССР примерно с 1927 г. стало кодирование с перешифровкой одноразовой гаммой. Известный ученый К. Шеннон впоследствии доказал, что такой шифр в принципе недешифруем[28].

VI. Конец 1920-х годов – вторая половина 1950-х годов

Начало эпохи «машинного шифрования», советские криптографы сумели обеспечить секретность военных, государственных и дипломатических сообщений в ходе Великой Отечественной войны. Активный вклад наших криптоаналитиков в достижение Великой Победы.

Уже в годы Первой мировой войны появились первые предложения по полной автоматизации шифрования и расшифрования. Наступала эпоха машинных шифров. В межвоенный период на смену ручным шифрам стали приходить шифромашины (механические и электромеханические устройства для шифрования)[29].

В СССР пионерами машинного шифрования стали специалисты, работавшие в области криптографической защиты речевого сигнала. Еще в 1920 г. русский ученый М.А. Бонч-Бруевич усовершенствовал временную перестановку, введя кадровую структуру преобразований, когда каждые N сегментов переставлялись по-своему. (Суть этого засекречивающего преобразования проста. Представим себе, что ваша речь записана на магнитную ленту. Эта лента разрезается на мелкие фрагменты, которые затем «склеиваются» по заранее заданному закону перестановки «отрезков». В этом склеенном виде информация поступает в канал телефонной связи. На приемном конце, зная правило перестановки, восстанавливается исходное сообщение.)

Первые разработки аппаратов секретного телефонирования в СССР относятся к 1927–1928 гг., когда в НИИС РККА были изго-

товлены для погранохраны и войск ОГПУ 6 аппаратов ГЭС (конструктор Н.Г. Суэтин)[30]. В 1930-х годах в области секретной телефонии вели работы 7 организаций: НИИ НКПиТ, НИИС РККА, завод им. Коминтерна, завод «Красная Заря», НИИ связи и телемеханики ВМФ, НИИ № 20 НКЭП, лаборатория НКВД. В 1931 г. была создана первая отдельная сеть междугородной высокочастотной связи (ВЧ-связь) с применением специальных средств защиты. В 1934 г. на заводе «Красная Заря» (Ленинград) начался крупносерийный выпуск трехканальной аппаратуры высокочастотного телефонирования СМТ-34, работающей в диапазоне 10,4–38,4 кГц и обеспечивающей удовлетворительное качество связи на расстоянии до 2000 км[31]. В 1935–1936 гг. на заводе «Красная Заря» было создано устройство автоматического засекречивания телефонных переговоров – инвертор ЕС (К.П. Егоров и Г.В. Старицын) и налажен его выпуск для каналов телефонной высокочастотной связи. Через год завод наладил выпуск шифратора ЕС-2 и его модификаций ЕС-2М, МЕС, МЕС-2А, МЕС-2АЖ, ПЖ-8М, к 1940 г. – выпустил 262 аппарата. Принцип работы новинки был достаточно прост: инверсия с одновременной подачей в канал связи мешающего тона с высоким тембром. Этими установками в 193 г. было оборудовано 9 междугородных правительственных линий связи, а на 1 апреля 1941 г. – 66 линий из имевшихся 134. Устройства типа «ЕС» успешно использовались для организации ВЧ-связи практически на всем протяжении Великой Отечественной войны и позднее. В лаборатории завода им. Коминтерна также до войны было разработано 4 типа аппаратов, в том числе СУ-1 с динамической инверсией и перестановкой двух полос спектра и система СЭТ-2 с динамической перестановкой 3-х полос спектра. Однако еще в 1940 г. констатировалось, что «разработанная по заказу НКВД заводом “Красная Заря” аппаратура для засекречивания телефонных разговоров обладает слабой стойкостью и не имеет кода»[32]. В 1939 г. В.А. Котельникову было поручено создание шифратора для засекречивания речевых сигналов с повышенной стойкостью. Заказчиком аппаратуры был отдел правительственной ВЧ-связи. В специальной лаборатории ЦНИИС была предложена система, основанная на квазислучайных (известных только получателю) перестановках временных (100 миллисекунд) отрезков и 2-частотных полос с инверсией речевого сигнала. Управление частотными и временными перестановками на передаче и приеме осуществлялось шифратором, генерировавшим 5 бит гаммы 10 раз в секунду. Разработка шифратора имела оборонное значение, и для ее завершения лаборатория во время войны была эвакуирована в Уфу, где ее сотрудники объединились с группой специалистов с завода «Красная Заря» из Ленинграда. Шифратор был создан к осени 1942 г.[33]

Во время Великой Отечественной войны разработанная под руководством В.А. Котельникова и испытанная еще в 1938 г., сложная засекречивающая аппаратура С-1 «Соболь» широко использовалась в действующей армии. Она применялась также для связи с Москвой нашей делегацией во время принятия капитуляции Германии в мае 1945 г. За создание аппаратуры засекречивания речи группе разработчиков и В.А. Котельникову в 1943 и 1946 гг. были присуждены Сталинские премии I степени. В лаборатории В.А. Котельникова проводились также исследования возможности создания аппаратуры засекречивания с использованием принципа полосного вокодера с выделением основного тона речи, открытого в 1939 г. американским инженером Г. Дадли. Работа была доведена до действующего макета, который был испытан и продемонстрировал возможность использования этого принципа для сжатия речевого сигнала. В ходе работы В.А. Котельников также предложил и опробовал принцип артикуляционного тестирования систем передачи речи. В 1941 г. он доказал, что можно создать математически недешифруемую систему засекречивания, если каждый знак сообщения будет засекречиваться выбираемым равновероятно знаком гаммы (совершенно стойкий, по К. Шеннону, шифр). Такая система должна быть цифровой, а преобразование аналогового сигнала в цифровую форму должно основываться на теореме отсчетов В.А. Котельникова. Такая аппаратура начала создаваться только после войны.

Аппаратура для шифрования текстовых сообщений появилась несколько позднее. В 1937 г. на ленинградском заводе № 209 им. А.А. Кулакова были произведены опытные экземпляры первого советского шифратора «В-4» (конструктор И.П. Волосок), реализующего шифр гаммирования[34]. В 1938 г. на заводе началось серийное производство данных шифраторов. В 1939 г. В.М. Шарыгиным была проведена модернизация шифратора «В-4», новая машина получила название «Изумруд» и стала производиться параллельно с «В-4» начиная с 1940 г.[35] В том же 1937 г. на заводе № 209 под руководством В.Н. Рытова был создан макет дискового шифратора. В 1939 г. эта шифромашина под названием К-37 «Кристалл» была запущена в серийное производство. В 1940–1941 гг. она выпускалась в Ленинграде, а в 1942–1945 гг. – на заводе № 707 в Свердловске. Выпуск машины продолжался до 1946 г.[36] К началу Великой Отечественной войны на вооружение шифрорганов СССР было принято более 150 комплектов К-37. Эта техника позволила в 5–6 раз повысить скорость обработки шифротелеграмм, при этом сохраняя стойкость передаваемых сообщений[37]. Во втором квартале 1939 г. на заводе № 209 были изготовлены опытные образцы аппаратуры засекречивания телеграфных сообщений «С-308» (для телеграфного аппа-

рата Бодо) и «С-309» (для отечественного телеграфного аппарата СТ-35). В третьем квартале 1939 г. начался серийный выпуск этой аппаратуры на заводе № 209, а в 1942–1945 гг. аппаратура производилась на заводе № 707 в Свердловске. В 1940 г. конструктором П.А. Судаковым был разработан военный буквопечатающий старт-стопный телеграфный аппарат со съемным шифрующим блоком «НТ-20». С января 1941 г. началось серийное производство данной аппаратуры на заводе № 209, а в 1942–1945 гг. эти шифромашины, как и другие упомянутые выше шифраторы, производились на заводе № 707 в Свердловске. Боевое крещение советские шифромашины получили в 1939 г. – аппаратура «В-4» использовалась в районе боевых действий у реки Халхин-Гол[38].

Советские дешифровальщики достигли значительных успехов во время военных конфликтов второй половины 1930-х годов. Так, во время гражданской войны в Испании на стороне республиканцев работали советские криптоаналитики[39]. В начале 1938 г. группа советских специалистов-дешифровальщиков была направлена в Китай. В течение следующих 19 месяцев советские криптоаналитики вскрыли 10 японских шифров, применявшихся армией и ВВС Японии в Китае. Советские специалисты вскрывали приблизительно 200 японских шифросообщений каждый месяц[40]. Успехов добились наши специалисты и во время событий на Халхин-Голе. Они поставляли нашему военному руководству важную информацию. Эта информация позволила изменить ход военных действий в нашу пользу.

Во время Великой Отечественной войны советские шифровальные службы обеспечили секретность наших сообщений, не позволили противнику получить сведения о наших замыслах и действиях, «советская шифровально-кодировочная аппаратура в военный период сыграла особую роль, поскольку именно ее использовали на важнейших направлениях скрытой связи. Именно она обеспечивала возможность оперативного закрытия важнейшей стратегической и оперативно-стратегической информации от противника»[41].

Приведем еще одну цитату о роли советской шифртехники во Второй мировой войне: «Созданная в предвоенные годы отечественная шифровальная техника в процессе Великой Отечественной войны держала свой первый по-настоящему серьезный экзамен на зрелость. Огромные ресурсы были вложены в эту войну, длившуюся четыре долгих года, и вместе со страной криптографическая военная служба прошла столь же непростой путь от поражений в начале войны до решающей победоносной фазы, повернувшей врага вспять. Советская криптография, сумевшая скрыть от врага наши стратегические планы, но раскрывшая многие намерения врага, внесла в

Победу свой весомый вклад»[42]. Надо отметить, что ни одна советская шифромашина не была взломана противником, хорошей стойкостью отличалась и значительная часть наших ручных шифров.

Вот как оценивали работу советских шифровальщиков прославленные полководцы Великой Отечественной. Г.К. Жуков: «Хорошая работа шифровальщиков помогла выиграть не одно сражение»[43], А.М. Василевский: «Ни одно донесение о готовящихся военно-стратегических операциях нашей армии не стало достоянием фашистских разведок»[44].

Оценили надежность наших шифров и представители противника. Так, начальник штаба при ставке верховного главнокомандования немецких вооруженных сил генерал-полковник А. Йодль в своих показаниях на допросе 17 июня 1945 г. сообщил: «Основную массу разведданных о ходе войны – 90 процентов – составляли материалы радиоразведки и опросы военнопленных. Радиоразведка – как активный перехват, так и дешифрование – играла особую роль в самом начале войны, но и до последнего времени не теряла своего значения. Правда, нам никогда не удавалось перехватить и расшифровать радиограммы вашей (советской. – *Д. Л.*) ставки, штабов фронтов и армий. Радиоразведка, как и все прочие виды разведок, ограничивалась только тактической зоной»[45]. А вот что говорил на одном из совещаний А. Гитлер: «Эти проклятые русские шифровальные машины, мы никак не можем их расколоть!»[46]

Очень активно работали советские радиоразведчики и криптоаналитики в ходе Великой Отечественной войны. Накануне войны наши дешифровальщики предупредили руководство страны о нападении Германии. В ходе войны советские дешифровальные службы предоставили политическому и военному руководству СССР большое количество важнейшей информации. Эта информация поступала во время всех важнейших сражений (битва за Москву, Сталинградская битва, сражение на Курской дуге и т. д.) и способствовала нашим победам. Советские криптоаналитики вскрывали и машинные шифры иностранных государств. В годы войны удалось дешифровать ряд немецких шифраторов (но не «Энигму»). Приведем оценку их работы, данную бывшим генеральным директором ФАПСИ генералом А.В. Старовойтовым: «Нам была доступна информация, циркулирующая в структурах Вермахта (почти вся!). Я полагаю, нашим маршалам была оказана существенная помощь в достижении перелома в ходе войны и, наконец, окончательной победы. Наши полевые центры дешифрования работали весьма успешно. Войну в эфире мы выиграли»[47]. А вот как оценивает деятельность советских специалистов в годы войны один из бывших руководителей советской радиоэлектронной разведки генерал-лей-

тенант П.С. Шмырев: «...Я часто вспоминаю Великую Отечественную войну. Помню себя и своих товарищей – радиоразведчиков 1941 года, когда мы мало знали и еще меньше умели. И вспоминаю их же и себя в 1943–1944 годах, когда радиооператоры знали по почерку чуть ли не всех немецких радистов, определяя по ним номера дивизий, корпусов, армий. Любая задача нашим радиоразведчикам была по плечу»[48].

Приведем ряд примеров успешной работы советских дешифровальщиков. Большой вклад советские дешифровальщики внесли в победу под Москвой, «...уже в первые дни войны Б.А. Аронским (с помощью своих помощников и переводчиков) были дешифрованы кодированные донесения послов ряда союзных Германии стран в Японии. По поручению императора Японии послы докладывали своим правительствам о том, что Япония уверена в их скорой победе над Россией, но пока сосредоточивает свои силы на юге Тихого океана против США (а ведь эта война тогда еще даже не началась!). Аналогичные сведения были получены С.С. Толстым путем дешифрования переписки линий связи высших эшелонов власти Японии»[49]. Чуть позже эта информация была подтверждена в донесениях знаменитого разведчика Р. Зорге. Таким образом, руководство СССР убедилось, что Япония в ближайшее время не нападет на нашу страну, и пошло на переброску войск с Дальнего Востока и из Сибири. Именно эти соединения сыграли решающую роль в ходе победоносного наступления.

В предвоенные годы Сергей Семенович Толстой возглавлял японский отдел дешифровальной службы НКВД. Одним из самых крупных успехов накануне войны было дешифрование группой специалистов во главе с Толстым японских шифромашин, известных под названиями, данными им американцами, «оранжевая», «красная» и «пурпурная»[50].

Накануне Курской битвы буквально за сутки до начала сражения наши криптоаналитики вскрыли шифрованный приказ Гитлера о наступлении. Перехватив радиограмму, связисты опознали почерк радиста ставки главнокомандующего противника, а по характеру передачи сделали вывод, что она содержит очень важный приказ. Дешифровальщики знали, что речь может идти о крупном наступлении, и предположили, что в конце документа находится подпись Адольфа Гитлера. С помощью атаки «открытый-шифрованный» текст криптограмма была раскрыта. Она подтвердила информацию из других источников, в том числе информацию из Великобритании и сообщения от нашего знаменитого разведчика Н. Кузнецова, назвавшего дату наступления немецких войск под Курском. Приказ Гитлера войскам гласил: «Этому наступлению

придается решающее значение. Оно должно завершиться быстрым и решающим успехом...»[51].

Для проведения операции на флангах Курского выступа были сосредоточены 50 отборных дивизий, 10000 орудий, 2700 танков и свыше 2000 самолетов. В дешифрованном приказе указывалось, что наступление начнется утром. Не верить этой информации было нельзя. Поэтому в 2 часа 20 минут советские войска начали артиллерийскую контрподготовку, которая причинила немцам, сосредоточенным на исходных рубежах, значительные потери. В ходе грандиозного сражения враг был разгромлен, потеряв большое количество живой силы и техники. Так, например, из-за больших потерь ВВС, понесенных под Курском, Германия вынуждена была впредь полностью отказаться от действий своей авиации по объектам нашего глубокого тыла.

В качестве еще одного примера приведем результаты деятельности дешифровально-разведывательной службы (ДРС) Главного морского штаба в годы Великой Отечественной войны 1941–1945 гг. Весьма незначительная по численному составу, не превышающему 150 человек, ДРС ВМФ СССР показала удивительную результативность и эффективность, выдавая непрерывно достоверные разведывательные данные самого разнообразного содержания, в том числе и стратегического значения. Объектами разведки стали не только собственно морские силы Германии и ее союзников, но также и приморские группировки сухопутных войск, и в первую очередь авиации – главной ударной силы немцев в войне против СССР. Всего морскими дешифровальщиками было вскрыто более 300 кодов и шифров Германии, ее союзников и нейтральных государств и прочитано несколько сот тысяч телеграмм. Вот какую оценку их деятельности давал летом 1942 г. Верховный главнокомандующий И.В. Сталин: «Если бы не было дешифровальной службы Черноморского флота, я не знал бы обстановки на Юге»[52]. Имелись серьезные достижения во время войны и у армейских дешифровальщиков и специалистов органов госбезопасности.

VII. Вторая половина 1950-х годов – 1989 г.

Со второй половины 1950-х годов при осуществлении криптографической деятельности в СССР начинают активно применяться электронно-вычислительные машины (ЭВМ), поступают новые образцы шифровальной техники различного назначения, созданы шифромашины нового поколения – электронные шифраторы с использованием интегральных микросхем[53]. Развивалась и укреп-

лялась отечественная радиоэлектронная разведка, «она в 50-е – первой половине 80-х годов прошлого столетия организационно и технически превратилась из фронтовой в стратегическую»[54]. Криптографическая деятельность в этот период является исключительной прерогативой государства.

VIII. 1989 г. – по настоящее время

В 1989 г. впервые в СССР открыто опубликован криптографический алгоритм – государственный стандарт ГОСТ 28147–89. В 1994 г. принимаются еще два криптографических стандарта – алгоритм цифровой подписи ГОСТ Р 34.10–94 (в 2001 г. вместо него появляется новый алгоритм цифровой подписи ГОСТ Р 34.10–2001) и хэш-функция ГОСТ Р 34.11–94. Криптографическая деятельность перестает быть монополией государства, с начала 1990-х годов на территории России ее начинают осуществлять коммерческие структуры, финансовые организации и частные лица. Появляются частные фирмы – производители шифровального оборудования, других средств защиты информации, а также специализирующиеся на оказании услуг в области защиты информации. Для регламентации осуществления криптографической деятельности, а также введения определенных ограничений на отдельные ее виды для негосударственных организаций принимается ряд законодательных и других нормативных актов (в качестве примера можно привести Закон РФ «Об информации, информатизации и защите информации»). Нормативно-правовая база в области защиты информации продолжает совершенствоваться и в настоящее время.

Примечания

[1] *Ларин Д.А.* Защита информации в Древней Руси // Вестник РГГУ. Серия «Информатика. Защита информации. Математика». № 12 (10). М.: РГГУ, 2010. С. 13–35.

[2] Соболева Т.А. История шифровального дела в России. М.: ОЛМА-ПРЕСС-Образование, 2002. С. 43.

[3] Там же. С. 44.

[4] *Астрахан В.И., Гусев В.В., Павлов В.В., Чернявский Б.Г.* Становление и развитие правительственной связи в России. Орел: ВИПС, 1996. С. 14.

[5] Очерки истории внешней разведки: В 5 т. / Под ред. Е.М. Примакова и С.Н. Лебедева. М.: Международные отношения, 1999. Т. 1. С. 37 (далее – Очерки).

[6] *Бабаш А.В., Шанкин Г.П.* История криптографии. Ч. I. М.: Гелиос, 2002. С. 208.

7 Очерки. Т. 1. С. 34.

8 Там же. С. 61.

9 *Астрахан В.И., Гусев В.В., Павлов В.В., Чернявский Б.Г.* Указ. соч. С. 14.

10 Там же. С. 16.

11 Там же.

12 Там же. С. 15–16.

13 Напомним, что криптоанализ – наука о дешифровании шифров, он применяется к «чужим» шифрам для получения информации и к собственным для оценки их стойкости и, соответственно, возможности использования для защиты своих секретов.

14 *Соболева Т.А.* Указ. соч. С. 73.

15 *Бабаш А.В., Шанкин Г.П.* Указ. соч. С. 215.

16 *Астрахан В.И., Гусев В.В., Павлов В.В., Чернявский Б.Г.* Указ. соч. С. 18–19.

17 Там же. С. 20.

18 Подробнее о работе российских криптографов в эпоху наполеоновских войн можно прочитать в статье: Ларин Д.А. Защита информации в эпоху Наполеона // Вестник РГГУ. 2009. № 10. М.: РГГУ, 2009. С. 10–32.

19 П.Л. Шиллинг (1786–1837) в течение ряда лет возглавлял цифирную экспедицию (шифровальную службу) МИД Российской империи. Он является изобретателем оригинального биграммного шифра, который более 80 лет использовался для защиты российских государственных секретов.

20 Подробнее с развитием конфиденциальной связи в России во второй половине XIX – начале XX в. читатель может ознакомиться в статьях: *Гольев Ю.И., Ларин Д.А., Тришин А.Е., Шанкин Г.П.* Научно-технический прогресс и криптографическая деятельность в России XIX века // Защита информации. INSIDE. 2005. № 2. С. 67–75; *Гольев Ю.И., Ларин Д.А., Тришин А.Е., Шанкин Г.П.* Начало войны в эфире // Там же. № 3. С. 89–96.

21 Описание российских шифров этого времени можно, в частности, найти в статье: *Бабаш А.В., Гольев Ю.И., Ларин Д.А., Шанкин Г.П.* Криптографические идеи XIX века. Русская криптография // Защита информации. Конфидент. 2004. № 3. С. 90–96.

22 Описание этого устройства можно найти в статье: *Бабаш А.В., Гольев Ю.И., Ларин Д.А., Шанкин Г.П.* Криптографические идеи XIX века // Защита информации. Конфидент. 2004. № 1. С. 88–95; 2004. № 2. С. 92–96.

23 Подробнее об этом см.: *Гольев Ю.И., Ларин Д.А., Тришин А.Е., Шанкин Г.П.* Криптография: страницы истории тайных операций. М.: Гелиос АРВ, 2008.

24 *Гольев Ю.И., Ларин Д.А., Тришин А.Е., Шанкин Г.П.* Начало войны в эфире // Защита информации. INSIDE. 2005. № 3. С. 89–96.

25 Подробнее о криптографической деятельности революционеров в России можно прочитать в цикле статей: *Бабаш А.В., Гольев Ю.И., Ларин Д.А., Шанкин Г.П.* Шифры революционного подполья России XIX века // Защита информации. Конфидент. 2004. № 4. С. 82–87; *Гольев Ю.И., Ларин Д.А., Тришин А.Е., Шанкин Г.П.* Криптографическая деятельность революционеров в 20-х – 70-х годах XIX века

в России: успехи и неудачи // Защита информации. INSIDE. 2005. № 5. С. 90–96; *Гольев Ю.И., Ларин Д.А., Шанкин Г.П.* Криптографическая дея-тельность организаций «Земля и Воля» и «Народная Воля» в России в 1876–1881 годах // Там же. № 6. С. 80–87; *Они же.* Криптографическая деятельность революционеров в России. 1881–1887 годы: агония «Народной Воли» // Там же. 2006. № 2. С. 88–96; *Они же.* Криптографическая деятельность революционеров в России в 90-е годы XIX века // Там же. № 4. С. 84–91; *Они же.* Криптографическая деятельность революционеров в России. Полиция против революционеров // Там же. 2008. № 2. С. 86–96; *Они же.* Криптографическая деятельность революционеров в России на рубеже веков (1898–1900 годы) // Там же. № 4. С. 89–96.

26 Более подробную информацию о событиях, связанных с криптографической деятельностью во время Гражданской войны в России, можно получить из статьи: *Ларин Д.А.* Криптографическая служба в годы Гражданской войны в России // Проблемы отечественной истории: Сб. науч. статей. Вып. 11. М.: Изд-во РАГС, 2009. С. 73–96.

27 См.: *Гольев Ю.И., Ларин Д.А., Тришин А.Е., Шанкин Г.П.* Криптография: страницы истории...

28 *Соболева Т.А.* Указ. соч.

29 *Бабаш А.В., Гольев Ю.И., Ларин Д.А., Шанкин Г.П.* О развитии криптографии в XIX веке // Защита информации. Конфидент. 2003. № 5. С. 90–96.

30 См.: *Астрахан В.И., Павлов В.В., Чернега В.Г., Чернявский Б.Г.* Правительственная электросвязь в истории России. Часть I (1917–1945). М.: Наука, 2001.

31 См.: Технические средства безопасности. [Элекронный ресурс] // Сайт Свирского Ю.К. [М., 2010]. URL: http://uks.dol.ru (дата обращения: 20.12.2010).

32 См.: *Астрахан В.И., Кириллычев А.Н.* У истоков секретной телефонии [Элекронный ресурс] // Энциклопедия ламповой радиоаппаратуры, выпуск № 162. [Москва–Донецк, 2002]. URL: http://amradio.ru/issues/issue162.htm#ogl (дата обращения: 20.12.2010).

33 См.: *Быховский М.* Пионеры информационного века. История развития теории связи. М.: Техносфера, 2006.

34 *Дадуков Н.С., Репин Г.А., Скачков М.М., Филин Ю.П.* Советская шифровальная техника. Ленинградский период: 1935–1941. Ч. 2: Пролог // Защита информации. INSIDE. 2006. № 2. С. 83–87.

35 *Дадуков Н.С., Репин Г.А., Скачков М.М., Филин Ю.П.* Советская шифровальная техника. Ленинградский период: 1935–1941. Ч. 3: Комбинат техники особой секретности // Защита информации. INSIDE. 2006. № 3. С. 93–96.

36 *Дадуков Н.С., Репин Г.А., Скачков М.М., Филин Ю.П.* Советская шифровальная техника. Ленинградский период: 1935–1941. Ч. 4: Расширение номенклатуры шифровальной техники // Защита информации. INSIDE. 2006. № 4. С. 92–96.

37 *Андреев А.* Именно у нас в городе тайное становилось явным // Гривна. № 48 (412). Херсон: Херсонская городская типография, 2002. С. 28.

38 *Дадуков Н.С., Репин Г.А., Скачков М.М., Филин Ю.П.* Советская шифровальная техника. Ленинградский период: 1935–1941. Ч. 4. С. 92–96.

[39] Подробнее об этом можно прочитать в статье: *Ларин Д.А., Шанкин Г.П.* Криптографическая деятельность во время гражданской войны в Испании // Защита информации. INSIDE. 2008. № 1. С. 62–64.

[40] *Соболева Т.А.* Указ. соч. С. 448–449.

[41] *Дадуков Н.С., Репин Г.А., Скачков М.М., Филин Ю.П.* Советская шифровальная техника. Ленинградский период: 1935–1941. Ч. 5: Накануне // Защита информации. INSIDE. 2006. № 5. С. 76.

[42] *Дадуков Н.С., Репин Г.А., Скачков М.М., Филин Ю.П.* Советская шифровальная техника. Ленинградский период: 1935–1941. Ч. 6: Первый экзамен выдержан! // Защита информации. INSIDE. 2006. № 6. С. 86.

[43] См.: *Жуков Г.К.* Воспоминания и размышления. М.: АПН, 1971.

[44] См.: *Василевский А.* Дело всей жизни. М.: Политиздат, 1978. С. 203.

[45] *Йодль А.* «Война с Россией – это такая война, где знаешь как начать, но не знаешь чем она кончится» // Сигары Шееле для «Барона Дризена». М.: Издательский дом Гелеос, 2001. С. 89–106.

[46] *Дадуков Н.С., Репин Г.А., Скачков М.М., Филин Ю.П.* Советская шифровальная техника. Ленинградский период: 1935–1941. Ч. 6. С. 86.

[47] *Кузьмин Л.А.* Не забывать своих героев // Защита информации. Конфидент. 1998. № 1. С. 83–85.

[48] *Бурнусов И.* Мэтр радиоэлектронной разведки // Независимое военное обозрение. 2009. № 35. С. 15.

[49] См.: *Кузьмин Л.А.* Указ. соч.

[50] Там же.

[51] См.: *Жельников В.* Криптография от папируса до компьютера. М.: ABF, 1996.

[52] *Куличенко В.* Русские против «Энигмы» // Независимое военное обозрение. 2004. № 40. С. 7.

[53] Некоторую информацию о советской шифротехнике этого периода можно получить из книги: *Астрахан В.И., Гусев В.В., Павлов В.В., Чернявский Б.Г.* Становление и развитие правительственной связи в России. Орел: ВИПС, 1996, а также из статей: *Дадуков Н.С., Репин Г.А., Скачков М.М., Филин Ю.П.* Советская шифровальная техника. Ленинградский период: 1935–1941. Ч. 1: Истоки // Защита информации. INSIDE. 2006. № 1. С. 91–96; *Кузьмин Л.А.* ГУСС – этап в развитии советской криптографии // Защита информации. Конфидент. 1998. № 4. С. 89–94.

[54] *Бурнусов И.* Указ. соч.

КОНЦЕПЦИЯ

А.Е. Баранович

СЕМАНТИЧЕСКИЕ АСПЕКТЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ: КОНЦЕНТРАЦИЯ ЗНАНИЙ

С позиции информационно-эволюционного подхода продолжается исследование основных направлений информационной безопасности интеллектуальных систем. Основное внимание в настоящей работе сконцентрировано на направлении их защиты «от информации». Статья продолжает цикл работ посвященных семантико-прагматическим аспектам обеспечения информационной безопасности.

Ключевые слова: аксиология, знания, знаний концентрация, интеллектуальные системы, избыточность информации, информационная безопасность, семантика, семантические фильтры, телеология, угрозы безопасности.

Введение

Основным объектом предметной области специальности ВАК Министерства образования и науки РФ «Методы и системы защиты информации, информационная безопасность»[1] является *информация*[2], в отношении которой решается проблема обеспечения безопасности (предмет исследования). К сожалению, приходится констатировать, что ясного понимания естественно-научной сущности упомянутого явления в рассматриваемой области до настоящего времени фактически не наблюдается. Исходя из сущест-вующих нормативно-правовых актов, законодательно регулирующих предметную область информационной безопасности (ИБ)[3], под *информацией* (И.) понимается ее субъективно-прагматическая интерпретация, а именно «сведения, факты, знания» и т. п., т. е. вторичный по отношению к информационной составляющей объективной реальности (ОР) продукт социальной деятельности антропных интеллектуальных систем (ИС)[4]. Настоящая терми-

нологическая система зародилась в начале XX в. и в современных науках «об информации» получила наименование «журналистского этапа» ее использования (и исследования). «Кибернетический этап» зародился в конце 40-х годов XX в. и в общих чертах сформировался к 60–70 годам. Данному этапу человечество обязано появлением «Математической теории коммуникации» К. Шеннона (одной из первых «теорий информации») и становлению криптографии (криптоанализа и криптосинтеза как ее разделов) как фундаментальной (теоретической) и прикладной (технической) науки. Это, в свою очередь, привело к созданию весьма эффективных (по вполне определенным критериям и показателям качества) технических средств преобразования, передачи и защиты информации в человеко-машинных информационных системах.

В этот же период началось формирование предметных областей информатики, интеллектуальных систем и искусственного интеллекта, где последняя (принадлежащая в настоящее время к области *общей* и *теоретической информатики*[5]) явилась в свет хронологически ранее первой[6] (1957–1967 и 1949–1956 гг.[7]) на волне развития кибернетики и вычислительной техники.

Целесообразно подчеркнуть, что ни в «Математической теории коммуникации» К. Шеннона[8] (и ее проекциях в различные теории информации), ни в кибернетике[9], ни в информатике вопрос построения и исследования аксиоматико-терминологической системы *явления* информации (как и ее динамической реализации – информатизации и информирования) не ставился вовсе. Более того, сам факт дефиниции (номинации) явления и формирования его смыслового контекста (семантического образа) принципиальным и декларативным образом выносился за рамки теоретических моделей перечисленных предметных областей (см. работу К. Шеннона «Бандвагон»[10]). Так что все последующие попытки приписать несуществующие результаты в исследовании явления информации перечисленным дисциплинам нельзя отнести к существенным и достоверным (М. Мазур[11]).

Третий, настоящий этап развития наук *об информации* («information / computer science») начал формироваться в конце 80-х – начале 90-х годов XX в. и фактически совпал с периодом зарождения постнеклассического этапа развития современной науки. Данный этап оказался весьма тесно связан с изменением общенаучных взглядов на природу информации в объективной реальности (материи в исторической интерпретации), опираясь при этом на последние достигнутые результаты в области фундаментальных естественных наук и, прежде всего, физики (физической космологии макромира, с одной стороны, и квантово-вакуумной физики мик-

ромира – с другой) и конструктивной математики, обобщенные на завершающем этапе познания конца XX в. во вполне определенным образом оформленный философский концепт[12].

Настоящая смена общенаучной парадигмы в отношении предметной области «информационной безопасности» естественным образом влечет существенное изменение взглядов как на постановку общезначимых задач ее обеспечения, так и на методологию их решения. Изложению ряда вполне определенных *семантических аспектов* обеспечения ИБ ИС (как естественного, так и антропогенного характера) посвящена настоящая работа, развивающая ранее достигнутые результаты в исследуемой предметной области[13].

Семантические аспекты информационной безопасности

Аксиоматико-терминологическая систематизация и классификация предметной области информациологии[14] осуществлены в атрибутивно-ингредиентной концепции информации[15]. К числу существенных субъективных свойств информации (И.) отнесены ее *прагматические свойства*, ряд результатов задействования которых в механизмах обеспечения ИБ ИС изложен в авторских работах[16]. К специфическим объектам современной информационной инфраструктуры отнесены самообучаемые антропогенные ИС, угрозы ИБ в отношении которых реализуются по аналогии как с *доинтеллектуальными* информационно-техническими объектами, так и с субъектами – пользователями инфраструктурой. Там же изложены основы методологии защиты ИС «от информации» с использованием *аксиологических фильтров*, реализующих функции *численной оценки ценности* поступающей информации, отбора наиболее ценной и *отсеивания* (фильтрации) менее ценной (*бесполезной* или *вредной*) с использованием вполне определенных критериев.

Под *характеристическим атрибутом* И., именуемым в настоящей терминологической системе семантикой (И.), в вербальном контексте обычно понимают интегральную совокупность ее смысла и значения[17], возможно и содержания.

В последнем аспекте (согласно определению семантического отношения как отношения информации и объекта – источника информации) семантика информации отражает *объективно-содержательные* атрибуты объекта (информационных форм материальных систем ОР[18]), являясь в данном контексте независимой от взаимодействующего субъекта.

С позиции двух первых, формирование любой социальной информации, в том числе и естественно-научных знаний об ОР, есть процесс деятельности антропного сознания. В этом контексте объективная информация неотрывна от субъективной интерпретации исследователя, т. е. от прагматического отношения И. к субъекту. Тем более, если речь идет об этапах восприятия и распознавания И. ИС, формирования ее смысла (однокоренное с «мыслью») и значения и, далее, фиксации субъективной интерпретации в вербальной форме в подсистеме знаний (ПЗ) антропных ИС (АИС). Таким образом, в предметной области существования и семантической коммуникации ИС объективная семантика И. неотъемлема от ее *прагматической составляющей*.

В парадигме информационно-эволюционного подхода материальные системы (МС) ОР (в модели нелинейной динамики[19]) проходят три этапа эволюции, порождая на каждом из них собственный подкласс МС ОР. На этапе формирования актуальной Вселенной (от сингулярной точки Большого взрыва до формирования адронной материи) – подкласс физических систем неживой («косной» по В. Вернадскому[20]) природы. На следующем этапе эволюции возникают кибернетические системы – материальные системы естественного или искусственного происхождения, характеризуемые наличием механизмов энергоинформационного *адаптивного управления*[21] собственным существованием во внешней среде (*А. Б.*). Кибернетическая система *взаимодействует* с внешней средой и достигает поставленной *цели* в процессе этого взаимодействия[22]. Оба упомянутых подкласса относятся к классу доинтеллектуальных систем. И наконец, на третьем витке эволюции ОР возникают интеллектуальные системы как кибернетические системы, обладающие интеллектуальными свойствами, а именно способностью в процессе адаптивного управления собственным существованием во внешней среде оперировать информационными *моделями* объективной реальности. Последнее предполагает наличие вполне определенной подсистемы знаний и принятия решений, той или иной степени развития, включающей механизмы сенсориума, синтеза, анализа, хранения и преобразования моделей ОР (знаний различного уровня категоризации), а также механизмы выработки решений на управление ИС[23].

В рамках используемого подхода *объективная семантика* И. характеризует информационные формы существования МС ОР и взаимосвязана с формой, структурой и организацией МС. Соответственно, в модельной интерпретации речь идет о некоторой универсальной структурной («структуралистической») модели информации МС. В данном контексте любые МС тождественной массы

(на уровне «стандартной[24] модели мира») различаются структурной организацией (массы кварков), т. е. информацией (ее семантикой – «содержанием»).

В свою очередь, *семантика субъективная* (прагматическая) интерпретируется в рамках ИЭП как динамический информационный образ объективной семантики (информации МС «внешнего мира»), инициализированный в подсистеме знаний воспринимающей ИС. В результате, в модельной интерпретации, прагматическая семантика S_I информации I в ИС ζ в состоянии («момент времени») f есть динамически активизированный входящей информацией I элемент $Z^I_{\zeta f}$ подсистемы знаний Z_ζ ИС, т. е. фактически некоторый оператор (в частном случае функционал) от *двух* (!) *переменных* $SI \equiv F\ (I, Z_{\zeta f})$. Последнее выражение применимо в отношении модельной интерпретации *любых прагматических свойств информации I*, характеризуемых собственным оператором F и *парой взаимосвязанных параметров* $(I, Z_{\zeta f})$[25].

В условиях согласованного развития двух основных направлений обеспечения ИБ ИС, а именно защищенности индивидуальных и коллективных информационных ресурсов интеллектуальных систем от *деструктивного внешнего воздействия* и от *несанкционированного их использования* (доступа), рассмотрение семантических аспектов обеспечения информационной безопасности начнем с направления защиты ИС от И.

Защита «от информации»: концентрация знаний

Как отмечалось в работах[26], экспоненциальный рост объема информации в коллективных информационных ресурсах и, в частности, в сети Интернет в условиях вполне определенных количественных ограничений на возможности средств ее восприятия, хранения, передачи и преобразования формирует новый класс угроз информационной безопасности, характеризуемых избыточностью совокупного входящего информационного трафика ИС.

Объемы активно предлагаемой (навязываемой) информации в Интернете существенно превышают возможности большинства интеллектуальных систем по ее осмысленной обработке (по одним оценкам, в 2009 г. количество[27] цифровой информации, хранящейся в компьютерных сетях, уже превысило 5 эксабайт[28], по другим – совокупный объем И., произведенной социумом в 2010 г., превысит 900 эксабайт)[29]. Элементарный ЕЯЗО-запрос в поисковой системе (браузере) Интернета влечет представление пользователю множества (от нескольких десятков до десятков тысяч) контекстно

близких документов, проанализировать которые в рамках реальных временных ограничений он не в состоянии. Использование 100 и более каналов (спутниковых цифровых) ТВ-вещания предполагает фактическое ограничение времени визуального контроля за выделенным каналом 14,5 и менее минутами в сутки (в среднем)[31].

Избыточность трафика влечет рост диффузии («рассеивания») полезной информации в открытом информационном пространстве. Информация как бы растворяется в потоке в лучшем случае бесполезной, а в худшем – вредной информации, в так называемом информационном мусоре, информационный объем которого зачастую на порядки превышает объем необходимой и полезной информации. В результате переполнение информационных ресурсов избыточной (бесполезной или вредной) информацией может повлечь за собой ситуацию, именуемую за рубежом «аналитическим параличом». Вышеупомянутые угрозы реализуются в условиях фактически неограниченной (в правовом аспекте) свободы распространения информации, вне ее семантико-прагматических качеств истинности (ложности, иллюзорности) и социальной ценности для индивидуума[32].

Общая методология защиты ИС от бесполезной информации с использованием разработанных автором методов, моделей и способов ее аксиологической («*ценностной*») фильтрации достаточно подробно изложена в работах[33]. Следует заметить, что в предложенной методологии в качестве объекта фильтрации определена не информация как таковая (в той или иной форме представления), но ее *семантика*, точнее вполне определенная формализованная *модель-универсум семантики*.

Рассмотрим еще один, на первый взгляд кажущийся очевидным (но только на первый), подход к решению актуальной проблемы борьбы с «информационным взрывом» в социальных информационных ресурсах (в частности, в Интернете). Подход, который может быть кратко охарактеризован выражением «концентрация знаний».

Сущность предлагаемого подхода заключается в управляемой эволюции[34] «информационной реальности» социума[35], характеризуемой использованием механизмов существенного сокращения объема циркулирующей в нем И. в целом (отвечающего динамике полиномиального или, более того, линейного роста И. во времени – в противовес неуправляемому экспоненциальному[36]).

В феноменологической основе подхода задействованы следующие прагматические свойства И. (ПСИ)[37].

1. *Избыточность* И, отражающая уровень превышения необходимого (минимально полного) для использования объема И. В понятие объема И. вкладываются как объективные характеристики количества И. (например, по К. Шеннону), так и ее субъективно-

прагматические параметры, отражающие содержательные (семантические) аспекты И.

2. *Краткость* И., характеризующая уровень сокращения объема используемой И. в отношении некоторой вполне определенной реперной единицы. В этом смысле краткость И. есть антоним ее избыточности.

3. *Кумулятивность* И., отражающая особенности эффективного функционирования индивидуальных и коллективных систем знаний АИС и заключающаяся в использовании в практической деятельности (семантической коммуникации) кратких (сжатых) форм представления И. При этом полную (в рамках ограничений системы знаний) И. о конкретном явлении можно восстановить (при выполнении условий целостности и доступности) по ее краткой форме представления. В частности, доказанное ранее утверждение на практике можно использовать без доказательства.

При вербальном представлении (номинации) знаний свойство кумулятивности И. реализуется путем категориальной свертки понятий, основанной на систематизации и классификации знаний[38].

4. *Рассеиваемость* И. Социальная информация способна рассредоточиваться по различным источникам. Одна и та же информация может быть представлена в различной форме в газете, журнале, книге, отчете, СМИ (радиовещание, телевидение) и т. д.[39]

В процессе управляемой эволюции (развитии) социальной «информационной реальности» реализуется ряд социальных антропогенно-технических механизмов сокращения производства, воспроизводства (размножения) и рассеивания бесполезной и вредной (ложной) информации, а также механизмов концентрации-аккумуляции семантической информации[40] в ограниченном информационном объеме (концентрация знаний в информационном обществе). С количественной точки зрения, речь идет *о сведении экспоненциального темпа роста* И. *к линейно-полиномиальному*, что согласуется с ростом ресурсных возможностей человечества по использованию информации (совокупная интеллектуально-вычислительная и коммуникационная мощность социума).

В данной постановке, в практическом плане происходит согласование информационных потребностей общества с его информационными возможностями. Причем с точки зрения коэффициента загрузки (формула Литтла) технологических средств обработки информации выполняется основное условие отсутствия перегрузки («паралича») систем массового (информационного) обслуживания: $\rho \leqslant 1$.

Последнее обстоятельство ставит вполне закономерный вопрос об определенной мифологичности *актуальной необходимости*

«квантовых вычислителей»[41], гипотетически реализующих некоторые дискретные алгоритмы экспоненциальной операционной сложности (для ДМТ[42]) за полиномиальное время. Тем более что и «квантовые вычислители» не в состоянии обеспечить приемлемые характеристики решения «в лоб» («жадными» алгоритмами) задач *гиперэкспоненциальной* сложности (весьма редко упоминаемых в общей теории NP-трудных задач). К ним, например, относится ряд алгебраических задач на k-гиперпространстве СХ-гипертопографов (моделирующих семантику информации) операционной (и/или емкостной) сложности ~ $O(2^{2^{\cdot^{\cdot^{2^n}}}})\}$ (k экземпляров 2)[43]. Это, однако, не исключает возможности их практического решения с использованием ряда предметно-ориентированных методов[44].

Приведем ряд наглядных примеров. Последствия «информационного взрыва» наблюдаются далеко не во всех предметных областях социума, но, прежде всего, в областях, связанных с массовой коммуникацией, СМИ, рекламой и т. п., занимающихся в основном размножением (рассеиванием) и интерпретацией информации. Поэтому на запрос в коллективный информационный ресурс о конкретной информации по конкретному событию, однозначно отражающему реальное состояние некоторой МС ОР, например, извержении крупного вулкана, вместо одной исчерпывающе обобщающей справки специалиста мы получаем сотни, тысячи условно семантически корректных и безусловно избыточных интерпретаций. Вместо одного информационного агентства (ИТАР-ТАСС), в крайнем случае нескольких (для повышения достоверности И.), мы имеем десятки и сотни. При этом запрос об общей теории алгебраических систем даст нам всего несколько авторских первоисточников и десяток зачастую сомнительных интерпретаций. То же самое касается и физики, и химии, и биологии, и техники. Тем самым подтверждается *тезис* о том, что *реальные общественные знания*, в частности в области естественно-научных дисциплин, *никоим образом не характеризуются экспоненциальным темпом роста* их информационного объема[45]. Скорее здесь наблюдается феномен включения в социальный информационный обмен массы индивидуумов (АИС), воспроизводящих информацию (вместо себе подобных) согласно закону неуправляемого экспоненциального роста биологической популяции[46] и не обладающих соответствующей профессиональной компетенцией в хаотически («псевдосвободно») формируемом социальном информационном пространстве[47].

Поучителен и пример использования информации в системах социального, в частности государственного, управления (особенно его высшего звена), где также не наблюдается ее экспоненциаль-

ного роста, прежде всего, вследствие концентрации социальных механизмов принятия решений в руках субъектов (АИС) социума (ЛПР – лиц, принимающих решения), обладающих весьма ограниченными возможностями по «аналитико-синтетической переработке документальных источников информации». В результате вместо пространных (перегруженных несущественной информацией, избыточных) документов в системах социального управления массово используются формализованные документы ограниченного объема (в несколько страниц формата А4), содержащие управленческую информацию в аккумулированном (концентрированном) виде.

Заметим, что вследствие утверждения[48] о принципиальном различии подсистем знаний АИС в общем случае число различных семантических интерпретаций одной и той же информации может достигать числа представителей социума, что хорошо иллюстрируется примером единственности И. любого авторского творческого произведения, например фуги или хорала И.С. Баха, но совершенной множественностью ее интерпретации исполнителями. В данном контексте *каждый субъект* «свободного» социума априори может быть отнесен к классу *«творческих» производителей* (трансляторов-генераторов) социальной информации, обладающих *собственной уникальной интерпретацией любой информации*, что в трактовке ЕЯ «обыденного сознания» характеризуется вербальными выражениями «иметь свою точку зрения» или «иметь свой взгляд». Здесь, однако, (повторно) возникают вопросы о профессиональной компетенции «свободных» интерпретаторов в области интерпретируемой информации, а также *необходимости, полезности, достоверности и безызбыточности* интерпретированной информации для потребителей. При отрицательных ответах на вышеперечисленные вопросы в лучшем случае речь пойдет о субъектах – генераторах *информационного мусора* и его *рассеивании* в социальном информационном пространстве, в худшем же – о производителях ложной (вредоносной) информации.

Вследствие ограниченных возможностей АИС (как в качественном, так и в количественном отношении) по реализации процедур очистки настоящего социального информационного пространства от бесполезной, вредной и избыточной информации и семантической аккумуляции (концентрации) знаний, в качестве основных механизмов осуществления обозначенной концепции целесообразно определить антропогенно-технические ИС (АГИС), обеспечивающие как пассивную (в локальных информационно-коммуникационных узлах)[49], так и активную (в распределенном сетевом пространстве)[50] фильтрацию-концентрацию информации (знаний) с задействованием среды «агентов-мусорщиков» и «агентов-концентраторов».

Использование технологии разработки АГИС, основанной на «архитектуре, обусловленной моделированием»[51], предопределяет выбор в качестве центрального звена проектирования механизмов «фильтрации-концентрации» – универсальной семантической модели информации, позволяющей формализовать и алгоритмически описать вышеупомянутые процедуры в отношении самого широкого круга приложений.

В качестве абстрактной модели-универсума семантической информации предлагается использовать апробированную[52] модель *k*-гиперпространства семиотико-хроматических гипертопографов (СХ-ητ-графов) *Гs k*-го порядка топологизации множества-носителя, редуцируемую в измеримое метрическое хроматическое булево *k*-гиперпространство над *GF(2)* и позволяющую эффективно интерпретировать известные модели представления декларативных знаний, включая семантические сети (метасети), фреймы и популярные (на сегодняшний день) концептуальные структуры и онтологии (OWL)[53]. Алгебраизация модели *Гs* в форме одноосновной метаалгебры $A_{\Gamma s}$ обеспечивает возможность моделирования динамических процессов функционирования ИС. Представление СХ-ητ-графа в виде элемента измеримого метрического булевого пространства позволяет перейти к непосредственной алгоритмизации разработанных методик и реализации их на существующих средствах вычислительной техники. Использование «плавающего» интервала топологизации множества-носителя, наряду с использованием «нечетких» вычислений[54] и формальными практическими ограничениями, налагаемыми на мощности задействованных множеств, обеспечивает работоспособность модели на существующих средствах вычислительной техники.

Кажущаяся очевидность рассматриваемого подхода оказывается при этом связанной с весьма непростым и далеко не очевидным процессом смены массовых социальных ориентиров социума (индивидуальных АИС), в модельной интерпретации – с изменением его (их) пространства целей («ценностей»)[55]. Последнее обстоятельство связано с проблематикой предметных областей *телеологии* и *аксиологии*, упомянутых еще в трудах Аристотеля[56]. Не углубляясь в тематику упомянутых дисциплин, выходящую за рамки ограничений настоящей работы, отметим ряд принципиальных факторов, способствующих реализации изложенного подхода.

Широко распространенная в массовом сознании до настоящего времени постановка вопроса о *полном удовлетворении безгранично возрастающих информационных потребностей* социума, так же бессмысленна, феноменологически и мировоззренчески неверна и пагубна для него, как и ее материально-вещественные предтечи

эпохи соревнования *советской* модели социализма (коммунизма) с капитализмом Европы и США в XX в.[57]

Неоднократно проводимый рациональный анализ ресурсных возможностей социума и биосферы Земли уже давно показал невозможность обеспечения его массовых потребностей на уровне потребления наиболее развитых стран[58]. Таким образом, в фундаменте концепции «устойчивого развития» уже сейчас должны лежать понятия необходимости и возможности, а не потребности, что естественно влечет за собой введение разумных ограничений на темпы их роста как в вещественно-материальном, так и в информационном воплощении. Соответственно, понятие социальной свободы информации в данном контексте необходимо интерпретировать прежде всего как ответственность за ее создание, распространение и использование, в противовес широко распространенной анархической трактовке, тяготеющей к биологическому эволюционизму[59].

В области материально-вещественного производства социум уже давно столкнулся с вполне определенными естественными (вещественно-энергетическими) ограничениями его роста, оформленными, кстати, в форме регламентирующих правовых документов национального и международного уровня. Достаточно упомянуть, например, ограничения, связанные:

- с массовыми экологическими проблемами локального и глобального характера (нарушение / разрушение необходимых и достаточных условий существования физико-химической и биологической среды обитания социума);
- с проблемами изменения климата (аномальные климатические явления, Киотский протокол по ограничению выброса парниковых газов);
- с производством вполне определенной технологической продукции и исследованиями по ее разработке (ядерные расщепляющиеся материалы, опасные для биосферы физико-химические и биологические ингредиенты, ряд направлений генной инженерии и индустрии и т. п.);
- с отсутствием (неразвитостью) инфраструктуры вторичного использования и утилизации (воспроизводства) производимой товарной продукции.

Совершенно аналогичная ситуация возникает (существует) и в сфере информационной индустрии, где ограничения на темпы производства информационной продукции также связаны со специфическими экологическими проблемами («информационная экология») и проблемами реальных активных угроз безопасности социума в целом («информационные войны», «информационное противоборство» и т. п.).

На сегодняшний же день в области производства и распространения И. мы имеем вполне определенный порочный круг ее воспроизводства, когда экономически и стратификационно выгодное и целенаправленное создание новых носителей информации и средств ее преобразования, характеризуемых высокой серийностью и малой себестоимостью образцов массового производства[60], в совокупности с пресловутой «свободой ее распространения» порождает ничем не оправданный и не обоснованный рост массовых информационных потребностей социума. Утешением может служить лишь то обстоятельство, что сам социум еще не вышел из периода «младенчества» активной информационной фазы своего развития[61]. И продолжающееся противоборство страт за возврат сообщества АИС на путь «социальной эволюции» (охватывающей, прежде всего, период XVIII–XX вв. европейской истории) в сторону «сферы разума» (ноосферы по В. Вернадскому[62]), в противовес ему навязываемой «биологизации» (регрессирующей «социологизации»), еще не завершено[63].

Процедуры решения задач автоматизированной (автоматической) фильтрации / концентрации знаний с использованием АГИС, наряду с такими вышеупомянутыми ПСИ (и их формальными математическими моделями), как ценность (важность, полезность), избыточность И., краткость И., кумулятивность И. и объем (количество) И. (в частности, по К. Шеннону в последовательной семиотической модели Дж. фон Неймана), задействуют и следующие ПСИ, участвующие в формировании семантики И.[64]:

1. *Истинность* И. – свойство И. (ее способность) истинно отражать реальную информационную картину ОР. В АИКИ *истинность отражения* характеризуется соответствием информационного образа отображаемой МС ОР ее информационному прообразу (информационной форме существования МС ОР) с сохранением гомоморфизма отображения в ПЗ ИС (в частности, совпадением проекции образа на прообраз с частью прообраза). В модельной интерпретации речь идет о свойствах сюръективного отображения абстрактной экспликации модели информационной формы (ИФ) МС ОР на абстрактную экспликацию модели информационного образа (ИО) МС в ПЗ ИС (СХ-ητ-графы).

В соответствии с общефилософским понятием истины И. может быть *объективно истинной* (*абсолютно* и *относительно*), *не зависящей* от социальных *субъектов*, *интерсубъективной* (единой, общепринятой для множества субъектов социума) и *субъективной* (индивидуальной). Первичная, внесоциальная И. (И. ОР) изначально априорно истинна, однако ее формирование для использования социумом (в форме знаний) до настоящего времени реализуется

посредством задействования антропного интеллекта, что необходимым образом придает ей субъективный (интерсубъективный) и прагматический характер. Основным критерием истинности социальной И. является практика ее использования.

2. *Иллюзорность* И., характеризующая пограничные возможности сенсориума и подсистемы знаний ИС по адекватному восприятию / пониманию И. Иллюзорная И. есть *ложная* И., формируемая субъектом / объектом-преобразователем вне его сознательно-целенаправленной деятельности65. При формировании иллюзорной И. субъект уверен в ее истинности, хотя в реальности данная И. ложна и не адекватна ОР.

3. *Ложность* И. – свойство И ложно отражать реальную информационную картину ОР (в интерпретации АИКИ – информационный образ не соответствует своему прообразу). Антоним истинности И. ложная И., классифицируется на *иллюзорную информацию* и *дезинформацию*. И. ОР изначально априорно истинна, соответственно *свойство ложности* она может приобрести *только в результате ее преобразования субъектом*-объектом преобразователем. Вследствие формирования системы знаний социума субъектами-преобразователями объективной И. понятие ложности И. является безусловно прагматическим и характеризует подкласс социальной И. Основным критерием ложности И., как и ее истинности, является практика ее использования.

Соответственно, в процессе фильтрации / концентрации знаний АГИС формирует семантический образ входящей информации, проверяет ее на истинность / ложность (в рамках своих интеллектуальных возможностей) и выявляет полезную, бесполезную и вредную И.[66] В зависимости от уровня интеллектуальности АГИС принимает решения:

- выдать рекомендации по запрещению использования ложной и бесполезной И., а также по блокировке источников «загрязнения» информационной среды;
- уничтожить бесполезную и/или ложную информацию, очистив информационное пространство;
- сохранить ее («для потомков») с блокировкой текущего доступа и использования;
- уничтожить бесполезную и/или ложную информацию, активно блокируя источники «загрязнения», вплоть до информационной атаки на ресурсы источников;
- вступить с источниками дезинформации в семантическую дезинформационную игру, используя аппарат условных семантических ситуаций[67] и предварительно классифицировав ложную информацию на иллюзорную и дезинформацию.

В зависимости от различных постановок задач возможно формирование и иных стратегий их решения.

Выявив полезную информацию, АГИС приступает к процедуре «концентрации знаний», объединяя (путем задействования операции «слияния»[68] на метаалгебре $A_{\Gamma s}$) полезную входящую информацию с индивидуальной (или внешней, коллективной) подсистемой знаний. И далее осуществляет ее оптимизационную реструктуризацию (операция «реструктуризации» подпространства *Гs*) с использованием алгоритмов сокращения информационного объема знаний (в частности, путем объединения / поглощения их тождественных экземпляров и сокращения опровергнутых) и повышения эффективности их использования (в отношении, например, поиска и доступа).

Заключение

Вследствие вполне определенных ограничений на объем материала, ряд аспектов обеспечения защиты ИС от «деструктивной семантической» информации, в частности от активных средств разрушения семантики ПЗ ИС («семантические вирусы») и особенностей их реализации и обнаружения, остались вне рамок настоящего материала и требуют отдельного изложения.

Последнее касается и направлений обеспечения *защиты информации* в коллективной среде ИС, основанного на использовании семантических атрибутов И., а именно *криптосемантики*[69], в теоретической основе которой лежит класс формальных обратимых преобразований семантики засекречиваемой И., в историческом плане именуемых «семантическими шифрами», в отличие от классических «криптографических шифров» по К. Шеннону, определяемых на структурно-статистической модели множества открытых текстов и связанных с преобразованиями их формального семиотико-синтаксического представления в модели Дж. фон Неймана.

Список аббревиатур
АГИС – антропогенные интеллектуальные системы
АИКИ – атрибутивно-ингредиентная концепция информации
АИС – антропные интеллектуальные системы
ЕЯ – естественный язык
И. – информация
ИБ – информационная безопасность
ИО – информационный образ (ИФ МС)
ИС – интеллектуальные системы

ИФ – информационная форма (существования МС), прообраз ИО
МС – материальная система (ОР)
ОР – объективная реальность
ПЗ – подсистема знаний (ИС)
ПСИ – прагматические свойства информации
СХ-ητ-граф – семиотико-хроматический гипертопограф

Примечания

1 В сфере высшего профессионального образования, согласно существующим и перспективным Федеральным государственным образовательным стандартам нового поколения (проект), указанной специальности соответствует направление 090000 «Информационная безопасность», включающее, в частности, специальности 090104 (ранее 075400) «Комплексная защита объектов информатизации», 090301 «Компьютерная безопасность» и 090305 «Информационная безопасность автоматизированных систем», области профессиональной ориентации которых связаны с решением проблем обеспечения информационной безопасности и защиты информации в условиях существования вполне определенных угроз в информационной сфере.

2 См.: ВАК Министерства образования и науки РФ. Паспорта специальностей научных работников. [Электронный ресурс]. [М., 2010]. URL: http://vak.ed.gov.ru/ru/help_desk/ (дата обращения: 20.12.2010); ФГУ «НИИ РИНКЦЭ». База данных «VAK». Паспорт специальности 05.13.19. [Электронный ресурс]. [М., 2010]. URL: http://www.extech.ru/library/spravo/vak/vak_pasport/ne_abonent/add.php?kod=05.13.19&str=1&page=4&kod1=5 (дата обращения: 20.12.2010); Федеральные государственные образовательные стандарты высшего профессионального образования. [Электронный ресурс]. [М., 2010]. URL: http://www.edu.ru/db/portal/spe/index.htm (дата обращения: 20.12.2010); Проекты Федеральных государственных образовательных стандартов высшего профессионального образования нового поколения. [Электронный ресурс]. [М., 2010]. URL: http://mon.gov.ru/pro/fgos/vpo (дата обращения: 20.12.2010).

3 См.: Федеральный закон от 27 июля 2006 г. № 149-ФЗ «Об информации, информационных технологиях и о защите информации» / Рос. газ. Федер. вып. № 165 (4131). 2006. 29 июля.

4 Интеллектуальные антропные и антропогенные (технические) системы, наряду с классическими, информационными (автоматизированными), входят в настоящее время в основной перечень объектов защиты.

5 См.: ФГУ «НИИ РИНКЦЭ». База данных «VAK». Паспорт специальности 05.13.17. [Электронный ресурс]. [М., 2010]. URL: http://www.extech.ru/library/spravo/vak/vak_pasport/ne_abonent/add.php?kod=05.13.17&str=1&page=4&kod1=5 (дата обращения: 20.12.2010).

[6] См.: Wikipedia, the free encyclopedia. Informatics (academic field). [Электронный ресурс]. [М., 2010]. URL: http://en.wikipedia.org/wiki/Informatics_%28academic_field%29 (дата обращения: 20.12.2010).

[7] См.: *Тьюринг А.М.* Вычислительные машины и разум // Хофштадер Д., Деннет Д. Глаз разума. Самара: Бахрах-М, 2003; Wikipedia, the free encyclopedia. [Электронный ресурс]. [М., 2010]. URL: http://en.wikipedia.org/wiki/Artifical_Intelligence (дата обращения: 20.12.2010); Википедия, свободная энциклопедия. [Электронный ресурс]. [М., 2010]. URL: http://ru.wikipedia.org/wiki/Искусственный_интеллект (дата обращения: 20.12.2010).

[8] См.: *Shannon C.E.* A Mathematical Theory of Communication // Bell System Techn. 1948. № 3–4. J. 27; *Shannon C.E., Weaver W.A.* The Mathematical Theory of Communication. University of Illinois Press, Urbana, 1949 (Пер. в кн.: *Шеннон К.* Работы по теории информации и кибернетике. М.: Иностр. лит., 1963).

[9] См.: *Винер Н.* Кибернетика и общество. М.: Иностр. лит., 1958 (*Winner N.* The Human Use of Human Beings. Boston, Mass., 1950); *Винер Н.* Кибернетика. М.: Сов. радио, 1958 (*Winner N.* Cybernetics or Control and Communication in the Animal and the Machine. N.Y., Wiley, 1948); *Эшби У.Р.* Введение в кибернетику / Пер. с англ. под ред. В.А. Успенского; Предисл. А.Н. Колмогорова. М.: Иностр. лит., 1959 (*Ashby W.R.* An introduction to cybernetics. London: Chapman & Hall ltd, 1956); *Бриллюэн Л.* Наука и теория информации / Пер. с англ. А. Харкевича. М.: Физматгиз, 1960; *Бриллюэн Л.* Научная неопределенность и информация / Пер. с англ., под ред. И.В. Кузнецова. М.: Мир, 1966; *Couffignal L.* La Cybernetique. Paris, 1963.

[10] См.: *Шеннон К.* Работы по теории информации и кибернетике. М.: Иностр. лит., 1963.

[11] См.: *Мазур М.* Качественная теория информации: Пер. с польск. М.: Мир, 1974.

[12] См.: *Стёпин В.С.* Становление идеалов и норм постнеклассической науки // Проблемы методологии постнеклассической науки: Сб. ст. / Отв. ред. Е.А. Мамчур. М.: ИФРАН, 1992.

[13] См.: *Баранович А.Е.* Универсальный подход к структурному моделированию директивно-целевых информационных процессов. Автоматная модель интеллектуального процесса оценки ценности информации на Х-гиперграфах: Сб. статей. М.: ГШ ВС РФ, 1997; *Он же.* Структурное метамоделирование телеологических информационных процессов в интеллектуальных системах. М.: ГШ ВС РФ, 2002; *Он же.* Защита «от информации» как компонент информационной безопасности интеллектуальных систем: аксиологические WEB-фильтры // Тр. VIII Междунар. научн.-техн. конф. «Интеллектуальные системы» (AIS'08). М.: ФИЗМАТЛИТ, 2008; *Баранович А.А., Баранович А.Е., Лишин Н.А.* Исчисление ценности прагматической информации в интеллектуальной программной среде «Аксион» // Тр. XI национ. конф. по ИИ с междунар. участ. (КИИ-08). Т. 3. М.: Ленанд, 2008; *Баранович А.Е.* Прагматические аспекты информационной безопасности интеллектуальных систем // Вестник РГГУ. Сер. «Информатика. Защита информации. Математика». Вып. 10. М.:

РГГУ, 2009; *Баранович А.Е.* Дидактические материалы к специальному курсу «Введение в информациологию и ее специальные приложения». М.: РГГУ, 2010.

[14] Информациология [от *информация* + ...логия] – междисциплинарная наука о сущности информации, ее характеристических свойствах, формах представления (существования, изменения, взаимодействия) и моделирования (*авт.*). Синоним (факт.) – информология [от лат. *informo* + ... логия].

[15] См.: *Баранович А.Е.* Дидактические материалы к специальному курсу «Введение в информациологию и ее специальные приложения».

[16] См.: *Баранович А.Е.* Структурное метамоделирование телеологических информационных процессов в интеллектуальных системах; *Он же*. Защита «от информации» как компонент информационной безопасности интеллектуальных систем: аксиологические WEB-фильтры; *Он же*. Прагматические аспекты информационной безопасности интеллектуальных систем; *Он же*. Дидактические материалы к специальному курсу «Введение в информациологию и ее специальные приложения».

[17] В частности, по Г. Фреге.

[18] См.: *Баранович А.Е.* Дидактические материалы к специальному курсу «Введение в информациологию и ее специальные приложения».

[19] *Кадомцев Б.Б.* Динамика и информация. М.: Ред. журн. «Успехи физ. наук», 1997.

[20] *Вернадский В.И.* Размышления натуралиста: В 2 кн. Кн. 1: Пространство и время в неживой и живой природе. М.: Наука, 1975 (в 2 ч.: Ч. 1: Проблема времени, пространства и симметрии (1920–1942). Ч. 2: О жизненном (биологическом) времени (1931)).

[21] Управления с обратными связями.

[22] В основе данного утверждения лежит целевой принцип гомеостаза биологических кибернетических систем У. Эшби. В иной постановке, последнее свойство присуще телеологическим кибернетическим системам прежде всего антропогенного происхождения (в технике).

[23] *Вернадский В.И.* Указ. соч.

[24] «Кварковой».

[25] В отличие от *объективных* свойств И., моделируемых оператором (функционалом) от *одной переменной* вида $S_I \equiv F(I)$.

[26] См.: *Баранович А.Е.* Структурное метамоделирование телеологических информационных процессов в интеллектуальных системах; *Он же*. Защита «от информации» как компонент информационной безопасности интеллектуальных систем: аксиологические WEB-фильтры; *Он же*. Прагматические аспекты информационной безопасности интеллектуальных систем.

[27] Точнее, информационный объем.

[28] Иначе, «экзабайт» (10^{18}).

[29] См.: Прогноз всемирного роста объемов информации до 2010 года. [Электронный ресурс]. [М., 2010]. URL: http://erpnews.ru/doc1764.html#x_erp

(дата обращения: 20.12.2010).

30 Естественно языковый.

31 Формирование «мозаичного сознания» ИС есть один из путей *разрушения* концептуальной целостности ее подсистемы знаний («мировоззрения» в антропной интерпретации).

32 См.: *Баранович А.Е.* Дидактические материалы к специальному курсу «Введение в информациологию и ее специальные приложения».

33 См.: *Баранович А.Е.* Структурное метамоделирование телеологических информационных процессов в интеллектуальных системах; *Он же*. Основные элементы методологии дискретного метамоделирования процесса исчисления ценности информации в интеллектуальных системах // Тр. VI Междунар. научн.-техн. конф. «Интеллектуальные системы» (AIS'06). Т. 1. М.: ФИЗМАТЛИТ, 2006; *Баранович А.Е., Баранович А.А., Лишин Н.А.* Исчисление ценности прагматической информации в интеллектуальной программной среде «АКСИОН» // Тр. XI национ. конф. по искусственному интеллекту с междунар. участ. (КИИ-08). Т. 3. М.: Ленанд, 2008.

34 См.: *Baranovich A.E.* Concept of operated evolution of a natural language: problem statement / Proc. of the 12th Intern. Conf. «Speech and Computer» SPECOM'2007. Vol. 2. M.: MSLU, 2007.

35 См.: История информатики и философия информационной реальности: Учеб. пособие для вузов / [Юсупов Р.М. и др.]; под ред. Р.М. Юсупова, В.П. Котенко. М.: Акад. проект, 2007; *Баранович А.Е.* Дидактические материалы к специальному курсу «Введение в информациологию и ее специальные приложения».

36 Заметим, что настоящий неуправляемый экспоненциальный темп роста информации в социуме соответствует неуправляемому (вне любой системы ограничений, в частности, на используемые ресурсы) экспоненциальному темпу роста биологических популяций, т. е. законам неуправляемой биологической эволюции.

37 См. ссылку 35.

38 См.: *Баранович А.Е.* Структурное метамоделирование телеологических информационных процессов в интеллектуальных системах.

39 Упрощенный аналог понятия *диффузии* И. в пространстве-времени. Основывается на формализме эмпирического закона Б. Викери 1: n: n^2: n^3: ..., характеризующего отношение числа изданий, содержащих первичное семантическое ядро информации (первоисточники), к их последующей интерпретации (на различных уровнях – от научного до популярного).

40 Истинной, в рамках настоящей общенаучной парадигмы, и социально полезной.

41 См.: *Валиев К.А., Кокин А.А.* Квантовые компьютеры: надежды и реальность. Ижевск: НИЦ «Регулярная и хаотическая динамика», 2001; *Хренников А.Ю.* Введение в квантовую теорию информации. М.: ФИЗМАТЛИТ, 2008.

42 См.: *Ахо А., Хопкрофт Дж., Ульман Дж.* Построение и анализ вычислительных алгоритмов. М.: Мир, 1979.

43 См.: *Баранович А.Е.* Семиотико-хроматические гипертопографы. Введение в

аксиоматическую теорию: информационный аспект.

[44] См.: *Баранович А.Е.* Введение в предметно-ориентированные анализ, синтез и оптимизацию элементов архитектур потоковых систем обработки данных; *Он же.* Семиотико-хроматические гипертопографы. Введение в аксиоматическую теорию: информационный аспект.

[45] Мы сознательно не рассматриваем научно-популярные и образовательные социальные среды, где количество интерпретаций знаний достигает значительных величин, например, как в популярной и востребованной области информатики. При этом можно весьма определенно констатировать, что все принципиально новые идеи в этой области сконцентрированы всего лишь в нескольких первоисточниках, в то время как остальные источники представляют собой вполне прагматические интерпретации первых (на основе компиляции известных результатов, см., в частности, «Дидактические материалы к специальному курсу “Введение в информациологию и ее специальные приложения”, разд. 1.1), что в итоге лишь подтверждает вышевысказанный тезис.

[46] Здесь мы отметим, что в настоящее время реальные ресурсные ограничения биосферы с учетом роста индивидуальных потребностей АИС фактически разрушили механизм их неуправляемого биологического воспроизводства. Вместо экспоненциального роста числа биологических носителей мы имеем в лучшем случае линейный рост.

[47] См.: *Люксембург А., Симкин В.* Диктатура естествознания // Независимая газета. № 29. М., 1994.

[48] См.: *Баранович А.Е.* Структурное метамоделирование телеологических информационных процессов в интеллектуальных системах.

[49] Реализация функции пассивной защиты от агрессивной «внешней» информационной среды.

[50] Реализация функции активной защиты, включающая средства превентивного нападения (как задачи двойственной к задаче защиты) на информационные ресурсы агрессора (в общих рамках концепции «информационного противоборства»).

[51] См.: *Баранович А.Е., Баранович А.А, Кузнецова И.А., Мерзликин В.Г.* Моделирование процесса информационного сопровождения жизненного цикла биологической системы // Матер. 8-й междунар. науч.-техн. конф. «Кибернетика и высокие технологии XXI века» (C&T-2007). Воронеж: ВГУ, 2007.

[52] С 1999 г.

[53] См.: *Баранович А.Е.* Семиотико-хроматические гипертопографы. Введение в аксиоматическую теорию: информационный аспект; *Он же.* К-гиперпространство семиотико-хроматических гипертопографов как универсальная модель представления фактографических знаний // Матер. IX междунар. конф. «Интеллектуальные системы и компьютерные науки». Т. 1, ч. 1. М.: МГУ, 2006.

[54] Весьма семантически «размытое» наименование, вследствие реализуемости аппарата вычислений посредством вполне «четких» и однозначно интерпрети-

руемых классических алгебраических конструкций.

55 В рамках разработанной методологии численной оценки ценности И., определение (формирование, конструктивный синтез, выявление, выбор, адаптация) пространства целей ИС относится к ее базисным компонентам. Вне решения упомянутой подзадачи (дефиниции целей ИС) сама постановка задачи оценки ценности И. теряет смысл и становится *не определенной*. Более того, постановка задачи «защиты от информации» в условиях *директивного формирования* пространства целей, *противоречащего имманентным целям* ИС, автоматически реализует ее переход в «двойственную» задачу – «информационного нападения» на априори защищаемую ИС, ибо ведет к кардинальному переформированию семантики ПЗ ИС.

56 См.: Древнегреческая философия. От Платона до Аристотеля: сочинения: Пер. с др.-греч. Харьков: Фолио; М.: АСТ, 1999.

57 См.: *Веркор, Коронель.* Квота, или Сторонники изобилия / Пер. И. Эрбург // Веркор и Коронель, Перек Ж., Кюртис Ж.-Л., Ремакль А. Французские повести: Пер. с фр. / Сост. и вступ. ст. Ю.П. Уварова; Ил. В.Л. Гальдяева. М.: Правда, 1984. (Vercors et Coronel. Quota ou Les Plethoriens. Paris, 1966); *Кара-Мурза С.Г.* Манипуляция сознанием. М.: Алгоритм-ЭКСМО, 2006.

58 См.: *Кара-Мурза С.Г.* Потерянный разум. М.: Алгоритм, 2005.

59 Конституция Российской Федерации. Официальное издание. М.: Юрид. лит., Администрация Президента РФ, 1997. Ст. 29. «...П. 4. *Каждый имеет право свободно* искать, получать, передавать, *производить и распространять информацию* любым законным способом... П. 5. Гарантируется *свобода массовой информации...*».

60 Принципиальная особенность современных информационно-вещественных технологий.

В сфере вещественно-информационных технологий индустриальной и постиндустриальной стадий экономического развития стоимость НИОКР по созданию прототипа (образца) высокотехнологического изделия, например автомобиля или самолета, близка к общей стоимости серийного производства. Изготовление каждого серийного образца требует задействования значительных вещественно-энергетических ресурсов и, соответственно, существенных трудозатрат.

Иная ситуация складывается в области информационно-вещественных технологий («чисто информационных» технологий в природе не отмечается вследствие постулата АИКИ «о неразрывности информации от материального носителя»), где стоимость НИОКР может составлять 90–95% общей стоимости производства продукции. Например, при производстве программного обеспечения («software»), когда себестоимость серийного образца близка к весьма ограниченной стоимости носителя информации (затраты на ее копирование минимальны) и пределы серийности практически не ограничены ресурсным компонентом. В результате, в экономическом контексте, стоимость проведенных НИОКР «размывается» по стоимости произведенной серийной

продукции. В асимптотике при росте объема серии себестоимость образца будет стремиться к стоимости носителя. Близкая ситуация наблюдается и в области производства современных малоресурсных аппаратных компонент («hardware») информационно-коммуникационных технологий.

61 См.: *Бард А., Зодерквист Я.* Netократия: новая правящая элита и жизнь после капитализма. СПб.: Стокгольмская школа экономики в С.-Петербурге, 2004.

62 См.: *Вернадский В.И.* О научном мировоззрении // Научная мысль как планетное явление. М.: Наука, 1991 (первая публикация в «Вопросы философии и психологии». СПб, 1902).

63 См.: *Кара-Мурза С.Г.* Потерянный разум.

64 См.: *Баранович А.Е.* Дидактические материалы к специальному курсу «Введение в информациологию и ее специальные приложения».

65 В отличие от *дезинформации* как ложной И., целенаправленно и сознательно формируемой субъектом-преобразователем, имеющим ясное представление об истинной И. об исследуемом объекте.

66 Различение понятий «полезной» и «вредной» для ИС И. с позиции системно-эволюционного подхода основывается на *телеологической* характеризации исследуемых ИС.

67 См.: *Баранович А.Е.* Структурное метамоделирование телеологических информационных процессов в интеллектуальных системах.

68 Там же; *Баранович А.Е.* Семиотико-хроматические гипертопографы. Введение в аксиоматическую теорию: информационный аспект.

69 Авторский термин.

А.Е. Сатунина, Л.А. Сысоева

АНАЛИЗ МОДЕЛЕЙ ПЕРЕХОДА К СЕРВИС-ОРИЕНТИРОВАННОЙ АРХИТЕКТУРЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ВУЗА

Описаны ключевые начальные точки и сценарии перехода к сервис-ориентированной архитектуре (СОА) информационных систем. Обоснованы преимущества использования СОА в информационной системе университета. Приведена модель информационной системы университета на основе СОА.

Ключевые слова: сервис-ориентированная архитектура (СОА), архитектура информационных систем.

Введение

В последние годы в сфере разработки корпоративных информационных систем растет интерес к использованию сервис-ориентированной архитектуры (СОА), появление которой в значительной мере вызвано широким использованием процессной модели управления предприятием / учреждением и тенденцией к усилению взаимосвязи ИТ и бизнеса.

С точки зрения ИТ существует два основных подхода к созданию автоматизированных систем управления предприятием / учреждением. Первый – это внедрение единого комплексного решения от одного вендора, которое полностью или в достаточной степени удовлетворяет всем потребностям предприятия. Второй подход – применение композитной архитектуры, в которой функционал системы формируется за счет разнородных ИТ-приложений.

Внедрение комплексных ERP-решений, начавшееся в 1990-х годах, выявило ряд проблем монолитных решений:

– трудности в развитии и модификации ИТ-систем в условиях быстрого изменения бизнес-потребностей заказчиков и требований рынка;

– задачи автоматизации предприятий стали охватывать не только внутренние бизнес-процессы, но и внешние, поэтому создание единого комплексного решения вызывает технические сложности.

В результате все большее количество разработчиков стало рассматривать композитную архитектуру как наиболее отвечающую современным требованиям. Появившаяся несколько лет назад концепция СОА обозначила новый этап в развитии композитного подхода.

Новизна СОА заключается в следующем:

– новый подход к интеграции приложений на основе стандартов;
– новый подход к модификации и развитию функциональности информационных систем на основе использования более высокоуровневых компонент – сервисов;
– предоставляет возможность на основе ИТ-сервисов создавать новые бизнес-процессы и модернизировать существующие;
– реализовывает взаимосвязь СОА и технологий BPM.

Отличие сервисов от программных компонентов проявляется в следующих подходах:

– каждый сервис всегда ассоциируется с конкретной бизнес-функцией, в то время как компоненты относятся к технологической категории архитектуры программного обеспечения;
– сервисы могут быть реализованы в виде достаточно сложных программных комплексов, т. е. сервис отражает не способ его программной реализации, а только лишь способ его использования;
– формирование сервисов проводится с учетом потребностей бизнеса, а выделение программных компонентов было исключительно задачей разработчиков программного обеспечения.

Итак, можно рассматривать СОА как подход к интеграции бизнес-процессов и поддерживающей их ИТ-инфраструктуры в форме безопасных и стандартизированных компонентов (сервисов), которые можно использовать многократно и комбинировать для адаптации к изменяемым приоритетам бизнеса[1].

Внедрение сервис-ориентированной архитектуры дает новые возможности как для ИТ, так и для бизнеса. С позиций информационной стратегии СОА предоставляет следующие преимущества:

– гибкость и адаптируемость приложений за счет проектирования систем на основе стандартов и сервисов;
– сокращение времени разработки за счет повторного использования сервисов;
– способность к взаимодействию в гетерогенной среде;

– сокращение затрат на ИТ-инфраструктуру;
– улучшение согласованности ИТ-активов с бизнес-целями;
– повышение окупаемости благодаря уменьшенной стоимости реализации за счет внедрения стандартов, повторного использования и интеграции с внешними участниками бизнес-процессов.

Использование СОА дает следующие преимущества для бизнеса:
– сокращение времени адаптации к изменениям внешней среды;
– выявление неэффективных участков в бизнес-процессах;
– окращение стоимости и затрат на интеграцию ИТ-приложений;
– соответствие обеспечения бизнес-целей информационными ресурсами и технологиями;
– повышение независимости ИТ-инфраструктуры от изменений организационной структуры;
– реализация более эффективных процессов за счет стандартов и повторного использования ИТ-средств;
– обеспечение гибкости бизнес-деятельности как при внутреннем взаимодействии, так и при совместной деятельности с внешними участниками бизнес процессов.

Модели перехода к СОА

В случае если предприятие / учреждение приняло решение о целесообразности перехода к СОА, то необходимо выбрать соответствующий подход к реализации данной задачи. Многие ведущие компании, занимающиеся разработкой ПО для СОА, предлагают определенные подходы и методики осуществления перехода к СОА.

Компания IBM[2] для быстрого и эффективного перехода к СОА выделяет в организации три ключевые бизнес-ориентированные стартовые точки – люди, процессы, информация и две ключевые стартовые точки с ИТ-позиций – соединение и повторное использование. Организация может выбрать ту точку входа, которая наиболее готова к внедрению СОА, и сконцентрировать наибольшие усилия на ней, не игнорируя тем не менее остальные точки входа.

Стратегия внедрения СОА с точки входа, ориентированной на удовлетворение потребностей пользователей и их совместную деятельность (точка входа – люди), может помочь решить следующие задачи:
– повысить производительность и эффективность совместной работы сотрудников за счет их взаимодействия в контексте бизнес-процессов;

– унифицировать доступ к множеству приложений и источников информации;
– повысить возможности интеграции бизнес-процессов за счет описания логики потока работ независимо от применяемых сервисов и бизнес-логики.

Внедряя СОА с точки входа, ориентированной на бизнес-деятельность (точка входа – процессы), организация может оптимизировать процессы на предприятии, повысив их эффективность и гибкость, а также улучшить управление бизнес-процессами. Такая стратегия внедрения помогает:
– оптимизировать процессы;
– быстрее реагировать на изменения внешней среды;
– быстрее внедрять новые процессы;
– повысить координацию совместной деятельности с позиций процессов;
– повысить окупаемость инвестиций;
– обеспечить согласованность бизнес-целей и задач ИТ.

Внедряя СОА с точки входа, ориентированной на информационное обеспечение (точка входа – информация), организация может улучшить доступность и согласованность информации, возможность ее совместного использования. Данная стратегия внедрения позволяет:
– собрать данные и сделать их доступными;
– обеспечить доступ к источникам гетерогенных данных внутри и вне организации;
– отделить источники информации от приложений;
– снизить затраты, связанные с доступом к данным и их преобразованием;
– повысить адаптируемость организации за счет предоставления повторно используемых информационных сервисов.

Внедрение СОА с точки входа «повторное использование», которая ориентирована на информационные технологии, позволяет организации повторно использовать, расширять, улучшать и создавать рабочие процессы. Использование данной точки входа может помочь:
– создавать гибкие, сервис-ориентированные приложения;
– сокращать стоимость и уменьшать время разработки ИТ-приложений;
– уменьшать объем создаваемого нового кода;
– использовать существующие внешние сервисы вместо разработки собственных;
– устранить дублирование процессов.

Внедрение СОА с точки входа «соединение», которая также ориентирована на информационные технологии, предоставляет организации возможности по упрощению информационной среды для объединения людей, процессов и информации. Эта стратегия помогает:

- создавать защищенные, надежные и масштабируемые подключения внутри организации и вне ее;
- гарантировать передачу информации по различным протоколам внутри и вне организации;
- эффективно выполнять внутренние и внешние бизнес-процессы корпоративного уровня;
- формировать доверительные отношения между партнерами;
- реализовывать совместную работу пользователей независимо от каналов или устройств.

Для каждой точки входа предлагается сценарий или определенный подход к внедрению СОА (рис. 1).

Особенности ИТ-среды вуза

Деятельность современного вуза носит многопрофильный характер, который требует организации и управления разнообразным спектром информационных ресурсов и систем.

В качестве основных компонентов ИТ-среды вуза, которые определяют уровень информатизации образовательного учреждения, выступают: системы электронного управления образовательной деятельностью; электронный портал; электронные библиотеки; системы дистанционного обучения (СДО); системы электронного документооборота (СЭД); системы управления организационной деятельностью; системы управления экономической деятельностью; системы автоматизации производственно-хозяйственной деятельности; аналитические системы управления деятельностью вуза.

Специфика ИТ-среды вуза проявляется в следующих аспектах.

1. Практически всем вузам присуща фрагментарная, или «лоскутная», автоматизация по отдельным направлениям. Одна из причин – образовательному учреждению свойствен достаточно широкий спектр задач и видов деятельности (от производственно-хозяйственных процессов до ведения образовательного процесса на базе современных онлайн-технологий), который невозможно охватить одной корпоративной информационной системой. В вузах до сих пор сохраняется потребность в специализированных прикладных системах.

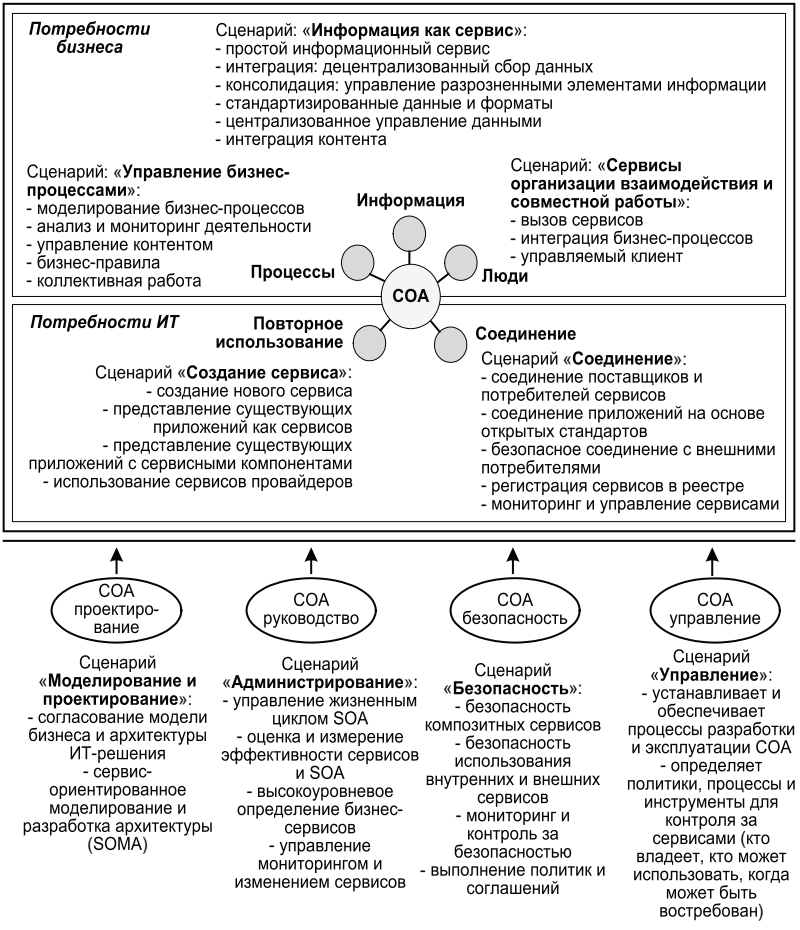


Рис. 1. Ключевые стартовые точки и сценарии для перехода к СОА

2. Образовательным учреждениям свойственна гетерогенная информационная среда, что связано с использованием значительного числа информационных систем и баз данных, созданных в разное время различными разработчиками на разнородных технологических платформах.

3. Уровень интеграции информационной среды невысок, и информация, хранящаяся в различных системах, либо частично дублируется, либо, наоборот, бывает неполной.

4. Ограниченные возможности в сфере финансирования развития ИТ-инфраструктуры. Данный факт являлся одной из причин того, что комплексные ERP-системы, появившиеся в середине 1990-х годов, практически не были востребованы в вузах, так как малое количество организаций могли себе позволить радикальное обновление программного обеспечения.

5. Низкий уровень использования аналитических систем поддержки принятия решений, одна из причин – отсутствие возможности получать консолидированные данные из нескольких источников.

6. Достаточно невысокая степень автоматизации в вузах собственно учебного процесса. Информатизация вуза в первую очередь затронула обеспечивающие подразделения (бухгалтерию, кадры, финансы и т. д.).

7. Образовательной сфере характерна динамичность как внутренней, так и внешней среды. В связи с этим от информационных систем требуется высокая гибкость и адаптируемость к изменяющимся требованиям.

8. Образовательная деятельность вузов всегда связана с инновационностью, что требует постоянного обновления ИТ-инфраструктуры вуза, а также включения и поддержки различных современных ИТ-технологий.

9. Невысокий уровень унификации бизнес-процессов образовательной деятельности. В каждом вузе они имеют свою специфику, что требует достаточно большой гибкости от ИТ-инфраструктуры вуза, чтобы соответствовать требованиям каждого образовательного учреждения.

Изменения в образовательной сфере, происходящие на федеральном, межвузовском и внутривузовском уровнях, выдвигают следующие требования к информационной системе образовательного учреждения:

- информационная система вуза в течение всего жизненного цикла должна быть изменяемой и адаптируемой к новым условиям и формам деятельности;
- новые компоненты информационной системы вуза должны легко и быстро интегрироваться в имеющуюся ИТ-инфраструктуру вуза;
- необходимо создать единое информационное пространство вуза, в котором эффективно взаимодействуют ранее внедренные и новые программные системы, независимо от их технологических платформ и средств разработки.

Одним из подходов к созданию современной информационной системы вуза, удовлетворяющей вышеперечисленным требовани-

ям, является использование методологии сервис-ориентированной архитектуры. Модель подхода к построению информационной системы вуза на основе СОА приведена на рис. 2. В такой модели ИТ-приложение или его часть является сервисом, который взаимосвязан с бизнес-процессом. В данном случае СОА можно рассматривать как концепцию создания и интеграции отдельных корпоративных приложений, задача которой – повысить гибкость корпоративной инфраструктуры, снизить затраты на разработку приложений и увеличить скорость реагирования на меняющиеся требования внешней среды.

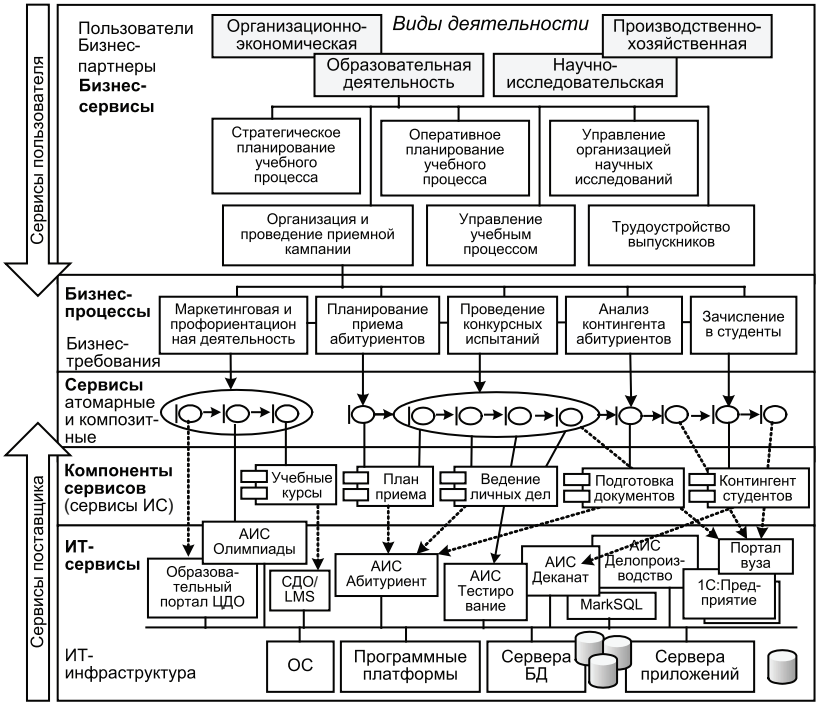


Рис. 2. Модель подхода к построению информационной системы вуза на основе СОА

Заключение

Использование сервис-ориентированной архитектуры для формирования современной композитной информационной системы вуза является одним из подходов к решению тех проблем, которые свойственны ИТ-среде вуза. СОА позволяет сохранить уже вложенные инвестиции и успешно функционирующие приложения, интегрировать их и облегчить расширение функциональных возможностей информационной системы за счет подключения новых сервисов.

Примечания

1 См.: *Биберштейн Н., Боуз С., Джонс К., Фиаммант М., Ша Р.* Компас в мире сервис-ориентированной архитектуры (SOA): ценность для бизнеса, планирование и план развития предприятия: Пер. с англ. М.: КУДИЦ-ПРЕСС, 2007. 256 с.

2 См.: *Beucker A., Ashley P., Borrett M., Lu Ming, Muppidi S., Readshaw N.* Understanding SOA Security. IBM International Technical Support Organization. 2007. P. 502.

А.С. Сысоев

ФОРМИРОВАНИЕ МЕТАДАННЫХ СЕРВИСОВ ДЛЯ ОЦЕНКИ ИХ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Рассмотрены вопросы информационной безопасности информационных систем, построенных на основе сервис-ориентированной архитектуры (СОА). Для определения уровней безопасности ИТ-сервисов используется метод формирования системы метаданных сервисов на основе декомпозиции на домены. Описаны параметры портфолио ИТ-сервиса.

Ключевые слова: сервис-ориентированная архитектура (СОА), архитектура информационной системы, информационная безопасность.

Введение

В информационных системах с сервис-ориентированной архитектурой (СОА) применяются те же основные составляющие информационной безопасности, что и для систем с традиционной архитектурой: конфиденциальность, целостность и доступность данных.

Тем не менее системам с СОА свойственны и специфические задачи в сфере безопасности:

– необходимость идентификации пользователей и сервисов и распространение этих данных в рамках всей организации и ИТ-инфраструктуры;
– необходимость легко соединяться с внешними ИТ-приложениями в режиме реального времени в процессе транзакций;
– необходимость обеспечения требуемого уровня безопасности для композитных приложений (поскольку для каждого сервиса могут использоваться различные уровни управления безопасностью, то требуется исследовать, как они будут влиять на уровень безопасности композитного сервиса);

– необходимость управления идентификацией и безопасностью в целом ряде систем и сервисов, которые вызываются в разнообразных сочетаниях и последовательностях и реализованы на основе новых и унаследованных приложений / технологий;
– необходимость соответствия системы безопасности постоянно возрастающему числу корпоративных и промышленных стандартов.

Для удовлетворения всех выше перечисленных требований необходимо, чтобы обеспечение и управление безопасностью в СОА охватывало все этапы жизненного цикла ИТ-приложения и строилось с учетом требований бизнеса и ИТ.

Система безопасности информационной системы с СОА

Значение безопасности становится еще более важным в ИТ-приложениях, построенных на основе СОА. В соответствии с принципами СОА система безопасности также должна отвечать требованиям динамичности и гибкой адаптивности к изменениям бизнес-процессов и внешней среды. Для реализации этих требований задачи безопасности должны быть учтены на каждой фазе жизненного цикла СОА-приложения.

Жизненный цикл ИС с СОА содержит четыре этапа (анализ требований и моделирование, сборка, развертывание, управление), которые образуют непрерывный замкнутый цикл (см. таблицу).

Таблица

Задачи безопасности, реализуемые в ходе жизненного цикла информационной системы с СОА

Этап ЖЦ	Описание
1. Анализ и моделирование	Анализ бизнес-деятельности и сбор требований. Моделирование и оптимизация бизнес-процессов. Разработка показателей для оценки эффективности
Задачи безопасности	Определение корпоративной политики безопасности. Идентификация требований к безопасности и ограничений. Выбор инструментов для сбора и анализа требований. Выявление и сбор требований к безопасности системы.

Продолжение таблицы

Этап ЖЦ	Описание
	Построение модели требований к обеспечению безопасности. Построение моделей безопасности приложений. Создание общей модели безопасности SOA-приложения
2. Сборка	Представление существующих приложений как сервисов. Создание новых сервисов. Определение внешних сервисов. Представление существующих приложений с сервисными компонентами. Тестирование сервисов. Сборка сервисов для реализации бизнес-процессов
Задачи безопасности	Формирование таблицы решений по безопасности для процесса внедрения SOA. Принятие и применение политик безопасности COA (возлагаются на приложения, сервисы безопасности COA-приложения, COA-компоненты). Декомпозиция требований по безопасности в набор сервисов безопасности. Создание инфраструктуры управления рисками. Выделение заинтересованных в безопасности COA-приложения лиц. Создание модели безопасности для каждого сервиса. Сравнение модели безопасности COA-приложения с моделями безопасностей сервисов. Построение и тестирование безопасности приложений. Обеспечение безопасности композитных сервисов. Обеспечение безопасности соединений
3. Развертывание	Настройка среды для реализации требуемого качества сервисов и удовлетворение системы безопасности. Масштабирование и оптимизация среды для повышения надежности реализации бизнес-процессов. Обеспечение гибкости процессов для динамического обновления
Задачи безопасности	Конфигурирование инфраструктуры для безопасности приложений. Настройка политик безопасности приложений. Интеграция людей, процессов и информации

Окончание таблицы

Этап ЖЦ	Описание
4. Управление	Мониторинг и управление сервисами, их доступностью и версионностью. Мониторинг ключевых показателей производительности (KPI) процессов. Диагностика проблем и определение направления совершенствования
Задачи безопасности	Управление безопасностью приложений. Управление идентификацией. Контролирование состояния приложения на соответствие политике безопасности. Управление политиками безопасности через множество приложений и инфраструктуру с целью соответствия изменяющимся требованиям. Контролирование поведения системы для выявления ситуаций, которые являются потенциальными угрозами безопасности, и внесение изменений в систему безопасности по мере необходимости. Анализ и оценка воздействия на бизнес определенных событий безопасности системы. Создание набора стандартов и интерфейсов прикладного программирования (API) для сторонних разработчиков

Компания IBM разработала эталонную модель безопасности в СОА[1], в которой выделяются два блока сервисов: сервисы безопасности бизнеса и сервисы ИТ-безопасности. Рассмотрим выполняемые ими роли.

Сервисы безопасности бизнеса включают управление потребностями и требованиями бизнеса, такими как управление идентификацией и доступом, защита данных, управление рисками и соблюдение политики, и могут быть классифицированы в шесть категорий (рис. 1).

В функции сервиса «соблюдение политики и отчетность» входят:

– соблюдение соответствия процесса управления функционированием бизнес/ИТ-систем мерам, предусмотренным политикой безопасности;
– проведение аудита информации обо всех событиях, а не только о событиях безопасности, и подготовка отчетности о поведении системы;

– использование данных аудита для оценки осуществления безопасности элементами SOA;
– выявление несоответствий между конфигурациями безопасности отдельных элементов ИТ-приложений и изменений в уровне безопасности в случае композитных сервисов;
– проведение проверки ИТ-приложений на соответствие внутрикорпоративным политикам безопасности, а также законам или нормативным актам.

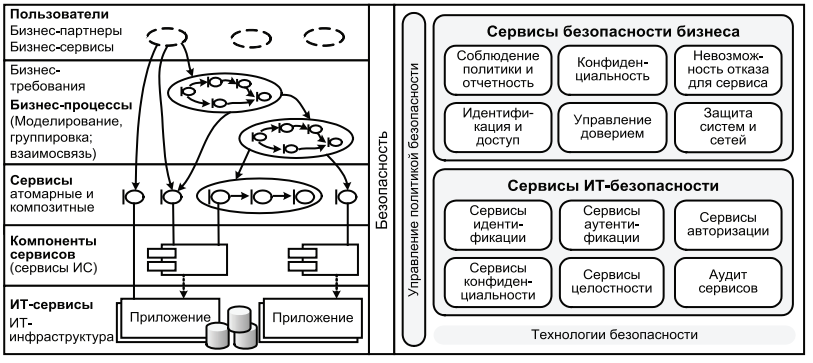


Рис. 1. Модель базовых сервисов системы безопасности в СОА

При наличии в ИТ-приложении информации с ограниченным доступом (персональные данные, финансовая информация и др.) должна быть обеспечена возможность ее защиты средствами политик доступа к данным, шифрования данных и поддержки безопасности приложений на аппаратном уровне, на уровне операционных систем и связующего программного обеспечения.

В функции сервиса «защита данных и управление конфиденциальностью» входят:
– формирование политики конфиденциальности на основе бизнес-политики;
– контроль за конфиденциальностью информации;
– формирование и контроль за правилами внешней обработки данных;
– управление пользовательским поведением в соответствии с политикой конфиденциальности;
– получение подробных отчетов о доступе к конфиденциальной информации и др.

«Невозможность отказа сервиса» используется для защиты потребителя и поставщика от ложных отказов в том, что данные /

запрос были отправлены или получены, т. е. отправитель не может отрицать, что он отправил сообщение, а приемник не может отрицать, что получил его. В настоящее время основой для осуществления сервиса безотказности является механизм цифровой подписи.

Сервис «идентификация и доступ» включает следующие функции:

- процессы и политику управления доступом к ИТ-ресурсам и бизнес-ресурсам;
- политику управления паролями и управление идентификационными данными;
- процессы, регламентирующие выполнение определенных задач без участия администратора, например, саморегистрацию пользователей, возможность изменить свои пароли и т. д.;
- обеспечение делегированного администрирования (делегирование действий другому пользователю или пользователям);
- управление политиками доступа к ресурсам на основе персональной идентификационной информации и информационных ресурсов;
- периодические проверки доступа к системам через определенные интервалы времени.

Сервисы идентификации и доступа применимы как на внутрикорпоративном, так и на межкорпоративном уровне.

Функционирование сервиса «управление доверием» направлено на обеспечение доверительных отношений между организациями, предприятиями, сферами безопасности, ИТ-системами. Эти отношения могут быть типа «система–система», «бизнес–бизнес». Управление доверием рассматривается с двух позиций: бизнеса и ИТ-технологий.

Доверительные отношения строятся на основе согласования взаимоотношений, формирования правил управления отношениями и мерами ответственности, описания бизнес-процессов и политик, которые необходимы для создания доверительных отношений, например политик доступа к ресурсам.

Сервис «защита систем и сетей» позволяет сформировать бизнес-политики, необходимые для обнаружения несанкционированных вторжений и управления событиями для обеспечения безопасности систем и сетей. В функции сервиса входит определение категорий технологий и встроенных систем, которые помогают защищать инфраструктуру серверов, систем и сетевых ресурсов от внешних и внутренних угроз безопасности.

ИТ-сервисы безопасности могут использоваться различными компонентами СОА, например прокси-серверами, серверами приложений, серверами баз данных и операционными системами.

Сервисы идентификации обеспечивают:

– управление, обмен, соединение и доступ к идентификационной информации, поступающей из различных источников, в том числе из нескольких систем идентификации;
– работу с несколькими реестрами пользователей;
– управление, хранение и использование информации об организационной структуре (пользователи, группы, роли);
– синхронизацию идентификационной информации через реестры пользователей.

Сервисы аутентификации обеспечивают возможность аутентификации как потребителя, так и поставщика сервиса. Эти сервисы могут поддерживать несколько механизмов аутентификации (имя пользователя / пароль, маркеры безопасности, биометрические параметры и др.). В СОА важную роль играет реализация SSO.

После успешного выполнения аутентификации запускается сервис авторизации. В СОА принятие решения о разрешении доступа к ресурсу для объектов зависит от двух аспектов:
– политики авторизации, описывающей необходимые атрибуты безопасности пользователя или системы, которая позволит им получить доступ к ресурсу;
– аутентификации пользователя или системы и их списка атрибутов безопасности.

Сервисы конфиденциальности обеспечивают защиту информации при передаче ее через коммуникационные сети или размещении в хранилищах данных. Информация, подлежащая защите, включает сведения о системе безопасности, пользователях, ИТ-приложениях, например криптографические ключи, пароли, персональную идентификационную информацию (PII) и др. Помимо шифрования дополнительными средствами обеспечения конфиденциальности данных и приложений являются средства аппаратного обеспечения, операционных систем и промежуточного программного обеспечения.

Сервис целостности используется для обнаружения несанкционированного изменения данных из-за ошибок или вредоносных атак. Организации должны позволять использование данных только авторизованным пользователям и приложениям, так же как и передачу данных для удаленной обработки.

Сервис аудита ведет подробный журнал для критических действий в бизнес-среде и ИТ-среде, которые могут быть связаны с несанкционированным доступом к защищенным ресурсам, например, изменение политики безопасности, несоблюдение указанной политики безопасности, изменение состояния безопасности серверов и т. д. Аудит должен проводиться для всех сервисов безопасности.

Метаданные сервисов для оценки информационной безопасности

Для формирования системы метаданных сервисов рассмотрим в первую очередь, какие группы параметров рекомендуется использовать для описания сервисов. OASIS (Организация по распространению открытых стандартов структурированной информации)[2] выделяет две группы параметров для описания сервиса (рис. 2):

1) описание динамики сервиса и взаимодействия с ним (используются параметры: обозримость, взаимодействие и эффект в реальном мире);

2) описание сервиса как объекта (сервиса как такового) (используются параметры: описание сервиса, контекст выполнения, контракт и политика).

Описание сервиса позволяет будущим потребителям принять решение о применимости сервиса к их потребностям, а также определить, удовлетворяют ли потребители требованиям поставщика сервиса.

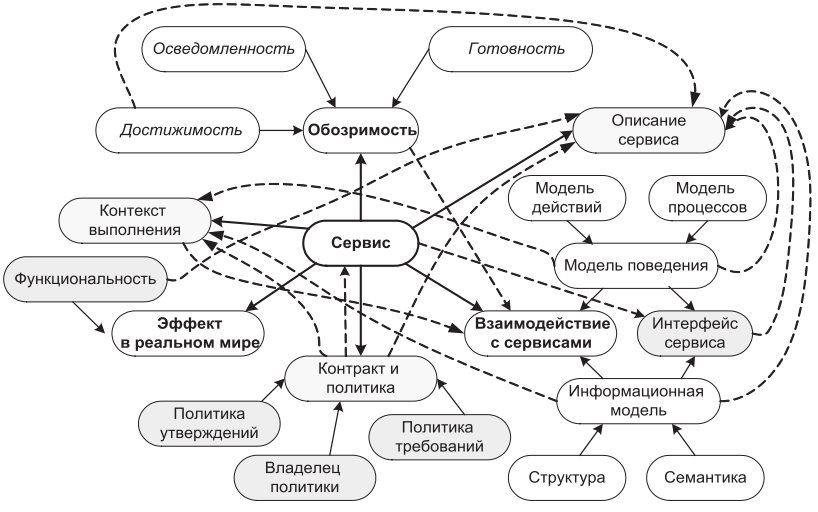


Рис. 2. Основные группы параметров описания сервиса

Методика формирования системы метаданных сервисов для оценки их безопасности строится на базе метода декомпозиции на основе доменов (рис. 3).

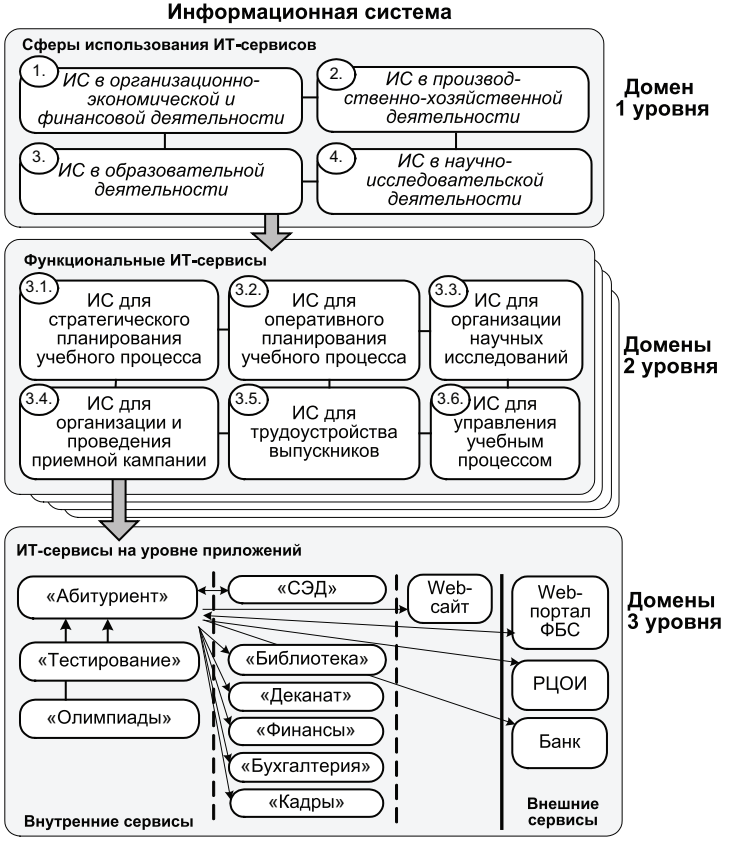


Рис. 3. Метод декомпозиции на основе доменов

Этап 1. Формирование домена первого уровня (бизнес-домены). Элементами домена являются основные направления деятельности предприятия / организации.

Этап 2. Формирование доменов второго уровня (функциональные домены). На основе декомпозиции выявляются функциональные ИТ-сервисы для каждого домена первого уровня (бизнес-домена). Элементами домена являются наборы ИТ-приложений, осуществляющие поддержку основных направлений деятельности предприятия / организации.

Этап 3. Формирование доменов третьего уровня (домены приложений). На основе дальнейшей декомпозиции выявляются ИТ-сервисы на уровне приложений для каждого домена второго уров-

ня. Элементами домена третьего уровня являются ИТ-приложения, реализующие конкретные функции.

Этап 4. Составление портфолио ИТ-сервисов. После того как сервисы домена выявлены, необходимо выполнить описание каждого ИТ-сервиса. Описание должно оставаться на достаточно высоком уровне, а детали будут определяться на стадии проектирования.

Портфолио ИТ-сервиса должно включать следующие параметры:

Параметр	Описание
Имя сервиса	Уникальное имя (идентификатор) сервиса
Аннотация сервиса	Краткое описание назначения сервиса
Ключевые слова	Ключевые слова, отражающие назначение сервиса
Область функционирования	Сфера деятельности сервиса. Определяется доменами первого и второго уровней
Собственник сервиса	Лицо / группа / организации, отвечающие за обслуживание сервиса
Классификация сервиса	Компонент сервиса: простой атомарный сервис, действующий на один ресурс, например, базу данных или программный код. Сервис для работы с данными: предоставляет данные по запросам и/или комбинации и преобразование данных из нескольких источников. Бизнес-сервис: атомарный сервис, состоящий из комбинации сервисных компонент и бизнес-правил. Workflow-сервис: длительный бизнес-процесс, который координирует действия различных сервисов и реализует внешние взаимодействия. ИТ-сервис: сервис на уровне ИТ-приложения
Форматы сообщений	Протоколы входящих и исходящих сообщений, включая служебные сообщения
Данные для взаимодействия	Расположение сервиса
Тип разработки	Промышленный (тиражируемый). Индивидуальный (заказной)
Производительность	Конкретные показатели производительности для сервиса (время обработки, количество одновременно работающих пользователей, время вызова и др.)

Функциональные параметры	
Функции сервиса	Текстовое описание функций сервиса, основные технические допущения
Число функций	Многофункциональный / однофункциональный
Результат выполнения сервиса	Эффект в реальном мире: сообщение (информация), действие и др.
Методы вызова сервиса	Описание методов вызова сервиса: для разных категорий пользователей могут быть предусмотрены разные варианты – через URL, локальный клиент
Нефункциональные параметры	
Качество выполнения	Определяет допустимую интенсивность отказов
Уровень доступности	Определяет допустимое количество и величину задержек сервиса при выполнении своих действий
Семантика	Определяет смысл терминов, используемых в описании сервиса и его интерфейсах
Способность к восстановлению	Определяет механизмы восстановления сервиса после различных видов сбоев
Аудит	Определяет виды и средства проведения аудита (активный аудит)
Протоколирование	Определяет виды и средства выполнения протоколирования событий (выборочное протоколирование)
Требования к безопасности	Обеспечение конфиденциальности, целостности, доступности сервиса
Уровень использования в СОА	ИТ-приложение будет входить в СОА. ИТ-приложение будет взаимодействовать с СОА-приложениями. ИТ-приложение не будет входить в СОА или взаимодействовать с СОА
Категории пользователей	Внутренние / внешние
Владелец сервиса	Внутренний (корпоративный) / внешний

Физическое размещение	Внутреннее / аутсорсинг / облачное
Территориальная распределенность	Централизованное хранение и обработка. Распределенное (механизм репликаций)
Механизмы тиражирования	Синхронное / асинхронное. Средствами сервиса / внешними средствами
Зона сети для размещения	Внешняя сеть. ДМЗ. Внутренняя сеть
Доступ через Интернет	Предусмотрен. Запрещен
Web-интерфейс	Предусмотрен / запрещен
Способ вызова в Интернете	URL / портал
Виды информации, с которой оперирует ИТ-сервис	Открытая информация. Информация со структурно-функциональным распределением доступа. Конфиденциальная информация
Выявление заинтересованных сторон в безопасности ИТ-сервиса	Внешние требования: организации по стандартизации и стандарты, которые формируют наборы специфических требований, клияющие на общую организацию безопасности СОА-приложений. Внутренние требования по безопасности: корпоративные стандарты безопасности последовательности операций
Уровень устойчивости бизнес-процессов	Классификация выполняется с учетом характеристик: уровень стабильности, коэффициент динамики изменений, коэффициент востребованности, коэффициент реагирования на инновационность ИТ-технологий, коэффициент старения
Механизм идентификации	Идентификационные данные. Организационная структура
Каталог пользователей	Каталог пользователей ИТ-приложения. Интегрированный каталог пользователей нескольких приложений. Общекорпоративный каталог пользователей

Управление паролями	Пользователь не имеет возможности изменить пароль. Пользователь имеет возможность смены пароля. Пользователь обязан регулярно изменять пароль в соответствии с политикой безопасности
Саморегистрация пользователей	Предусмотрена саморегистрация. Запрещена саморегистрация
Механизм аутентификации	Односторонняя / двусторонняя. Пароль, личная карточка, идентификационный номер, криптографический ключ, маркеры безопасности, биометрические параметры
Поддержка SSO	Поддерживается / не поддерживается
Использование шифрования данных при хранении данных	Шифрование на уровне документов (записей). Шифрование на уровне групп документов. Шифрование на уровне полей
Использование шифрования данных при передаче данных	Протоколы с шифрованием данных. Протоколы без шифрования данных
Использование шифрования для видов информации	Шифрование предметной конфиденциальной информации. Шифрование системной конфиденциальной информации
Управление доступом	Произвольное (дискреционное) управление. Принудительное (мандатное) управление Ограничивающий интерфейс. Ролевое управление
Механизм авторизации	Роли / группы
Механизмы контроля целостности	Целостность отдельного сообщения или поля информации. Целостновть потока сообщений или полей информации (порядковые номера, временные штампы, криптографическое связывание или иные аналогичные приемы
Поддержка ЭЦП	Не применяется. На уровне ИТ-сервиса. На межсервисном внутреннем уровне. На межсервисном внешнем уровне

Заключение

Информационная система современной организации является сложной, многофункциональной, многосервисной структурой, которая пользуется многочисленными внешними сервисами и предоставляет собственные сервисы вовне.

Специфика таких систем с точки зрения безопасности проявляется в том, что для каждого ИТ-сервиса основные аспекты информационной безопасности (доступность, целостность, конфиденциальность) трактуются по-своему и защита безопасности строится по-разному. Исходя из этого следует анализировать защищенность ИТ-сервисов.

Кроме того, системам, построенным на основе СОА, характерен переход от традиционных статических моделей безопасности к динамическим моделям, которые обеспечивали бы быструю адаптацию системы безопасности к изменениям в архитектуре и требованиям по безопасности. Для создания такой динамической модели безопасности необходимо на первом этапе разрабатывать и вести портфолио ИТ-сервисов, параметры которых будут использованы для формирования и анализа системы безопасности СОА-приложений.

Примечания

1 См.: *Beucker A., Ashley P., Borrett M., Lu Ming, Muppidi S., Readshaw N.* Understanding SOA Security. IBM International Technical Support Organization. 2007. P. 502.

2 См.: OASIS Reference Model for Service Oriented Architecture 1.0 / Committee Specification 1, 2 [Элекронный ресурс] // Новостной сайт CNews, версия для КПК. [М., 2006]. URL: http://www.pda.cnews.ru/reviews/index.shtml?2007/08/16/262770 (дата обращения: 20.12.2010).

М.И. Забежайло

К ЗАДАЧЕ МОДЕРНИЗАЦИИ КОМПЛЕКСА ИНФОРМАЦИОННЫХ СИСТЕМ КРУПНОГО КОММЕРЧЕСКОГО БАНКА

Предложена технология формализации и поиска решений для задач, возникающих в проектах модернизации ИТ-инфраструктуры российских коммерческих банков. Обсуждаются ее возможности и ограничения.

Ключевые слова: банковский бизнес, модернизация, ИТ-инфраструктура, методы формализации, банковские ИТ-проекты.

Многовековая история этого ремесла учит, что традиционное банковское дело – это, конечно же, *искусство* (и прежде всего искусство удовлетворения потребностей клиентов). Тем не менее быстро растущие в последние два-три десятилетия масштабы банковского бизнеса дают основания говорить об особой роли информационных технологий (ИТ) в его поддержке и сопровождении. Действительно, с точки зрения инфраструктуры крупный современный банк – это, в том числе, компьютерная «фабрика» массовой обработки огромных объемов данных о сделках с клиентами.

В подобной ситуации представляется естественным уточнить роль и границы применимости *методов* (в первую очередь с точки зрения классической дилеммы *искусство или метод*?) в части комплекса проблем, связанных с ИТ-сопровождением банковской деятельности. Одна из возможных целей такого уточнения – определить, в какой мере *искусство* ведения банковского бизнеса (в его современном состоянии) должно опираться на *методы* формирования и сопровождения ИТ-инфраструктуры такого бизнеса.

Рассмотрим более детально в предложенном контексте задачу формирования в крупных российских коммерческих банках эффективной ИТ-инфраструктуры. Сегодня вряд ли кто-либо из профессиональных экспертов будет оспаривать ее актуальность[1], а также

нетривиальный характер с точки зрения современных computer science & engineering. (В ряде российских банков в последние 7–10 лет были предприняты попытки модернизации ИТ-систем, однако ожидаемых позитивных результатов эти проекты, вообще говоря, принести не смогли[2]).

Следуя в предложенном направлении, представляется целесообразным попытаться сформировать *систематизированный* и *целостный взгляд* на эту предметную область – задачи модернизации банковских ИТ-систем (в том числе их *формализованное*[3] *описание*), важнейшими составными частями которого стали бы:

- обобщение как позитивного, так и негативного опыта уже выполненных банковских проектов ИТ-модернизации (в том числе выбор и использование в этом процессе удобной технологии представления знаний о рассматриваемой предметной области);
- выявление критически важных факторов, определяющих успех (или же неуспех) каждого соответствующего модернизационного проекта;
- формирование[4] своего рода «*мета-алгоритма*»[5] (детерминированной последовательности целенаправленных действий, обоснованных как с методической[6], так и с организационной[7] точки зрения), позволяющего устойчиво переходить от варьируемых исходных данных к планируемому в каждом конкретном случае позитивному финальному результату (построению целевой ИТ-системы);
- формирование набора аргументов, позволяющих обосновать его корректность (т. е. то, что он, будучи корректно исполняемым, всегда ведет к решению) и результативность (т. е. то, что полученное с его помощью решение будет удовлетворять исходным требованиям – иметь требуемые функциональные характеристики, порождаться в приемлемые сроки и в рамках приемлемых бюджетов и т. д.).

Говоря неформально, хотелось бы перейти (опираясь при этом на хорошо известные конструкции – научный метод, методику, алгоритм, эвристику и т. п.) к определенному уровню обобщения и типизации: подняться от *искусства* управления конкретным проектом банковской ИТ-модернизации к «*технологии*» (методике, методу) работы с *любым* проектом подобного типа. Как этого добиться?

Первым делом попробуем уточнить постановку решаемой задачи. Итак,

ЗАДАЧА:

Сформировать методику, а также комплекс реализующих ее исполнимых операций, обеспечивающих контролируемое (уп-

равляемое), надежное (устойчиво приводящее к планируемому результату), воспроизводимое (пригодное для многократного использования в варьируемых сопутствующих обстоятельствах) построение (в рамках соответствующего проекта модернизации ИТ-ландшафта) гибкой (в части предоставляемого пользователю набора финансовых продуктов и услуг) и масштабируемой (в части как реализуемой функциональности, так и наращиваемых объемов обрабатываемых данных) архитектуры (под которой понимается комплекс, объединяющий архитектуру процессов, архитектуру поддерживающих исполнение процессов программных приложений, а также обеспечивающую их работу системно-техническую архитектуру) полнофункциональной ИТ-системы универсального российского коммерческого банка.

При поиске решения данной задачи следует принять во внимание ряд специфических дополнительных обстоятельств, среди которых, разумеется, и растущие конкурентные потребности российской экономики, в том числе уже упомянутые ранее планы создания в ближайшей перспективе в РФ международного финансового центра. Однако при этом не следует забывать следующее:

1. В России, в отличие от ведущих западных стран-конкурентов (в части предоставления международных финансовых услуг и сервисов), нет столь же обширного опыта создания и эксплуатации подобных ИТ-систем. (Во времена СССР банковская система страны была ориентирована на нужды экономики другого типа – планового хозяйства. Таким образом, богатого опыта разработки и сопровождения ряда весьма актуальных в настоящий момент банковских услуг и сервисов[8] у российского банковского сообщества до последнего времени попросту не было.)

2. Российские стандарты бухгалтерского и налогового учета не позволяют напрямую (без доработок) использовать в ИТ-системах российских банков соответствующее зарубежное программное обеспечение (построенное на имеющих ряд существенных отличий учетных принципах).

Комплекс действий, призванный обеспечить *формирование* прикладного *ИТ-решения* в соответствии с зафиксированными поставленной ЗАДАЧЕЙ условиями, – это проект, понимаемый как целенаправленная упорядоченная последовательность операций (определенная часть которых, как правило, может выполняться и в параллельном режиме). «Область выбора» для таких операций (т. е., грубо говоря, – множество возможных вариантов для выбора допустимых действий) может быть очерчена (сформирована), например, в процессе обобщения опыта уже реализованных проектов модернизации банковских ИТ-систем.

Так, анализ накапленных данных о проектах рассматриваемого типа[9,10] показал, что общей группой элементов описания каждого из подобных проектов оказывается единая структура их *Жизненного Цикла* (ЖЦ) – последовательности проектных мероприятий (в некоторых случаях исполняемых и в параллельном режиме), характеризуемой существованием определенного и достаточно хорошо описанного множества проектных действий (допустимых проектных «активностей»), сопоставляемых каждому из шагов ЖЦ. Именно из такого множества вариантов выбирается конкретный набор исполняемых проектных «активностей» соответствующего шага ЖЦ. При этом на некоторых шагах ЖЦ подобное множество допустимых действий может формироваться всего из одного элемента, что соответствует исполнению (или же неисполнению) некоторого конкретного задания, например:

а) такого, как назначение конкретного лица на конкретную роль в проекте (это может быть, в частности, назначение определенного сотрудника на роль главного архитектора проекта и т. п.); или

б) фиксация в виде официального документа проекта тех или иных согласованных данных (в частности, это может быть соглашение о требованиях к качеству решений, разрабатываемых на том или ином конкретном этапе проектных работ, и т. п.).

Детальное представление о структуре ЖЦ проектов рассматриваемого нами типа дает *Дорожная Карта*[11] проектов комплексной модернизации ИТ-систем крупного российского коммерческого банка. Подробный анализ доступных «ответственному руководителю»[12] в подобных проектах вариантов выбора конкретных действий на конкретных шагах ЖЦ представлен в работах[13].

Таким образом, складываются основания говорить о множестве потенциально возможных *проектных траекторий*, каждая из которых задается соответствующей комбинацией выбираемых последовательно на соответствующих шагах ЖЦ проекта тех или иных допустимых вариантов действий. При этом очевидно, что все множество потенциально возможных проектных траекторий распадается на два подмножества: тех, что ведут к успеху проекта (будем называть их «позитивными»), и все оставшиеся (будем называть их «негативными»).

В подобной терминологии решаемая нами исходная ЗАДАЧА (см. выше) может быть переформулирована как задача *поиска* (для конкретного ИТ-проекта рассматриваемого типа) в соответствующем множестве всех возможных проектных траекторий по крайней мере одной «*позитивной*» *траектории*.

Проводимый параллельно анализ факторов, влияние которых способно препятствовать успешному проведению проектов рас-

сматриваемого типа к достижению поставленных целей[14], позволил сформировать соответствующую *Карту Рисков* (см. Приложение). Оказалось, что каждый из отдельных перечисленных в ней риск-факторов, вообще говоря, способен помешать успеху соответствующего проекта, т. е. он представляет собой своего рода СТОП-фактор в организации и проведении проектных работ.

В подобных условиях поиск подходящих средств противодействия влиянию каждого из выявленных СТОП-факторов стал необходимой составной частью работ по выявлению (идентификации описаний) таких проектных траекторий, которые позволяли бы приводить каждый из проектов рассматриваемого типа к успешному завершению – формированию целевого ИТ-решения в приемлемые сроки, оставаясь при этом в рамках приемлемых проектных бюджетов.

Опыт проводившихся банковских проектов ИТ-модернизации[15] (как и внеэмпирический анализ содержательной семантики риск-факторов, собранных в представленной *Карте Рисков*) показал, что далеко не все имеющиеся в распоряжении ЛПР в рамках реального проекта организационные и системно-технические средства *комбинируемы* между собой.

Так, например, в условиях *неполноты* требований к целевому решению (характерной для начальных стадий значительного числа крупных проектов рассматриваемого нами типа) использование при запуске работ и их последующей реализации классической технологии проектирования и разработки, т. е. организационной схемы:

проектирование => разработка => внедрение

вообще говоря, не ведет к успеху проекта[16]. (Соответствующие конкретные подтверждения легко найти в реальной практике выполнения модернизационных проектов рассматриваемого типа).

Таким образом, между группами факторов, попавшими в *Карту Рисков*, существуют нетривиальные[17] зависимости. И как следствие, актуальной оказывается задача *выявления* подобных *зависимостей*, так как только достаточно полное представление о них дает возможность найти *группируемые* (взаимно совместимые) варианты противодействия влиянию «атомарных» риск-факторов и в результате сформировать (возможно, не единственный) вариант искомой нами «позитивной» проектной траектории.

Изучение накопленных данных (причем одновременно и эмпирического опыта конкретных модернизационных ИТ-проектов, и доступной информации о семантике анализируемой ПО, в том

числе о природе существующих здесь риск-факторов, особенностях используемых информационных технологий[18] и т. п.) позволило:

1) выделить комплекс из шести взаимосвязанных групп риск-факторов, взаимодействие между которыми (в условиях использования тех или иных конкретных организационных и системно-технических решений как основы для управления соответствующими рисками) может оказать критически важное влияние на успех соответствующего проекта. Выявленный здесь комплекс связывает (как существенным образом взаимодействующие между собой):

(а) полноту требований к целевому решению, имеющихся в наличии на ранних стадиях проекта;

(б) тип архитектуры бизнес-процессов, требующих реализации средствами выбираемых (формируемых) прикладных программных решений;

(в) тип архитектуры программных приложений и выбор базовой SW[19]-платформы для реализации этих приложений;

(г) тип (используемая модель) технологии проектирования и разработки целевого ИТ-решения;

(д) тип (характер, модель) технологии управления проектом формирования целевого ИТ-решения;

(е) тип (способ, модель) контрактования исполнителей работ по проекту;

а также

2) предложить конкретный комплекс системно-технических и организационных решений, определяющих соответствующую «позитивную» траекторию Tr^+ со следующими конкретными характеристиками:

– компонентная[20] модель формализованного представления реализуемых бизнес-процессов в качестве основы для используемого по группе требований (б) решения;
– SOA[21] (а также подходящей промышленной SOA-ориентированной SW-платформы) в части используемого решения по группе требований (в);
– технология масштабирования решений-прототипов в качестве используемой технологии по группе требований (г);
– специальная проблемно-ориентированная комбинация технологий классического и экстремального управления проектами по группе требований (д);
– соответствующая модель контрактования исполнителей работ по проекту в части используемого по группе требований (е) решения.

При этом характеризуемый траекторией Tr+ комплекс (решение):

– позволяет успешно противодействовать влиянию выявленных (и перечисленных в Карте Рисков) СТОП-факторов ИТ-проектов рассматриваемого здесь типа;
– обеспечивает взаимную совместимость предлагаемых им для разных групп факторов риска мер противодействия их (т. е. СТОП-факторов) влиянию.

Дополнительно представляется важным отметить, что предложенное проектное решение имеет более широкую, чем изначально очерченная, область применимости22. Так, оказалось, что найденная комбинация определенных организационных и системно-технических мероприятий актуальна и в ряде предметных областей, находящихся за пределами собственно банковской сферы, например, везде в ИТ-системах управления российскими предприятиями, где:
– имеется финансово-налоговый блок (т. е. имеется необходимость комбинировать решения в рамках РСБУ и МСФО-учет);
– необходимо встраивание соответствующих систем информационной безопасности в формируемые ИТ-решения и др.

Кроме того, предложенная «позитивная» проектная траектория Tr^{+} предлагает новые (и достаточно элегантные) решения для ряда важных промежуточных задач, в том числе:
– в части встраивания отвечающих за обеспечение информационной безопасности решений (ИБ-решений) в формируемый целевой ИТ-ландшафт;
– для реализации так называемого «параллельного» учета[23] и др.

В части предоставляемых в распоряжение ЛПР возможностей для *объяснения* и *понимания* наблюдаемых в изучаемой предметной области явлений обратим внимание на специфичную (для решаемой ЗАДАЧИ – см. выше) цепочку «*шагов рассуждения*», сложившуюся в ходе формирования и развития обсуждаемого подхода:
(а) анализ имеющихся данных и, как результат, формирование абдуктивных[24] *объяснений* «негативности» / «позитивности» конкретных проектов-прецедентов, обусловленных (*комбинаторикой*[25] вложения / невложения в их формализованные описания) соответствующих *риск-факторов*;
(б) поиск средств (организационного и системно-технического характера) противодействия влиянию выявленных подобным путем риск-факторов;
(в) выявление неэффективности стратегии, так сказать, «пофакторного» противодействия, в том числе обнаружение «контрпримеров» (ряда случаев, когда такая стратегия, вообще говоря, не работает); выявление и фиксация новых фактов, говорящих о том, что не все возможные варианты

противодействия влиянию ряда конкретных риск-факторов оказываются комбинируемы (совместимы) между собой;

(г) вывод о том, что существуют нетривиальные (не являющиеся, например, уточняющими – см. выше) зависимости между соответствующими группами риск-факторов;

(д) заключение о том, что выявление и учет нетривиальных зависимостей дает дополнительные аргументы (*рациональные основания*) в пользу принятия (и использования в конкретных проектах) формализуемых в рамках развиваемого подхода приемов, т. е. применения таких проектных решений, которые:

– *соответствуют* имеющимся *фактам* (накапливаемым в рамках реализации соответствующих проектов эмпирическим данным)

и при этом оказываются

– *неоспоримыми*, т. е. в имеющейся описательной схеме[26] для них отсутствуют *фальсификаторы*[27], причем:

а) и в текущем контексте «семантических связей» между риск-факторами;

б) и в имеющихся эмпирических данных.

Таким образом, в рамках развиваемого здесь подхода можно вести речь о своего рода *рациональной аргументации* в части обоснованности принимаемых результатов (в ситуации, когда для построения *доказательств* попросту не хватает выразительных возможностей используемого языка представления знаний, а также достаточных объемов данных, накопленных в имеющемся эмпирическом материале).

* * *

Итак, с точки зрения стоящей перед ЛПР (в конкретном проекте рассматриваемого нами типа) задачи развиваемый подход:

– дает вполне работоспособные *средства представления и анализа знаний* об изучаемой предметной области, характеризуемые определенным уровнем *рациональности* (в общезначимом для современного исследовательского сообщества смысле этого понятия[28]);

– позволяет накапливать и развивать способствующие успеху (ЛПР и их проектов) знания о рассматриваемой ПО;

– позволяет ЛПР в ряде случаев *осознанно* и *систематически* (используя предложенный в рамках этого подхода *систематизированный* и определенным образом упорядоченный набор действий, обеспечивающих достижение поставленных целей) избегать «*слепого*» *перебора*. Другими словами, разви-

ваемый подход предоставляет возможности *целенаправленно* действовать в продвижении соответствующего проекта к успешному решению поставленных перед ЛПР задач.

Помимо прочего, развиваемый подход предоставляет ЛПР достаточно эффективный инструментарий управления бюджетными ресурсами[29], выделяемыми на реализацию подобных проектов.

Приложение. *Карта Рисков.*

I. Основные группы угроз (факторов риска)

1. *Условия общего характера*

1.1. «*Все сразу*» или «*по частям*»: **неадекватная стратегия**

Риски неудачного (по отношению к имеющемуся контексту конкретных условий, в которых предстоит исполнять необходимые изменения) выбора стратегии комплексной модернизации ИТ-ландшафта в рамках альтернативы «все сразу» или «по частям»?

1.2. Реализуемость **целевой функциональности**

Риски, связанные с целевой функциональностью, в том числе с адекватностью выбора платформенного ПО (в части целевой функциональности), адекватностью выбора целевой архитектуры, реализуемостью в рамках разумных ресурсных ограничений требований государственных регуляторов и др.

1.3. **Неполнота требований**

Риски, связанные с неполнотой требований к целевому решению на начальных стадиях его разработки.

1.4. **Функциональная гибкость**

Риски, связанные с необходимостью обеспечить функциональную гибкость целевой системы (см. также проблемно-ориентированные уточнения в разделах 2–4).

1.5. **Масштабируемость**

Риски, связанные с необходимостью обеспечить масштабируемость (как в части функциональности, так и в части объемов обрабатываемых данных) (см. также проблемно-ориентированные уточнения в разделах 2–4).

1.6. **Соответствие законодательству РФ**

Риски, связанные с необходимостью обеспечить соответствие требованиям законодательства РФ (см. также проблемно-ориентированные уточнения в разделах 2 и 3).

1.7. **Надежность**

Риски, связанные с необходимостью обеспечить надежность, в том числе:

– функциональную «разумность»[30] встроенных в целевую систему бизнес-процессов;
– функциональную полноту;
– необходимый уровень оттестированности программного обеспечения;
– поддерживаемость / сопровождаемость[31] формируемых решений и др.

1.8. **Защищенность**

Риски, связанные с необходимостью обеспечить защищенность (см также выше – соответствие требованиям законодательства РФ), в том числе в части:

– минимизации рисков реализации НДВ[32];
– надежной защиты банковской тайны;
– надежной защиты персональных данных и др.

(См. также проблемно-ориентированные уточнения в разделах 3 и 4).

1.9. **Экономичность**

Риски, связанные с необходимостью обеспечить экономичность, в том числе – в части:

– минимизации ТСО (общей стоимости владения) на сети вида ГО + филиалы;
– минимизации эксплуатационных расходов на сети[33] вида ГО + филиалы;
– минимизации расходов на сопровождение изменений.

1.10. **Процессно-реальное время**

Риски, связанные с необходимостью обеспечить оперативность функционирования целевого решения, в том числе процессно-реальное время принятия управленческих решений, и в подразделениях, и на сети вида ГО + филиалы.

1.11. **Неадекватные** проектные **решения** в условиях **неполноты требований**

Обусловленные неполнотой требований к целевому решению риски:

– неадекватной оценки ресурсов (сроков, бюджетов и профильных компетенций), необходимых для построения целевого решения;
– принятия неадекватного решения о форме контрактования исполнителей (например, решение о разработке собственными силами или принятие таких организационных решений, в которых оказываются приемлемы ситуации, где допустимы и некорректные (оставляющие возможности для действий, которые не ведут к успеху) условия взаимодействия с исполнителями проекта.

1.12. **Недостижимость целей выбранными средствами**

Риски недостижимости планируемых целей имеющимися в распоряжении средствами.

1.13. **Нештатное прерывание проекта**

Риски нештатного завершения / прерывания проекта внедрения целевой системы, управление которыми (в частности, их минимизация) требует в том числе:

- разделения рисков и ответственности за успех проекта между заказчиком и исполнителями (генеральным подрядчиком и субподрядчиками);
- выбора адекватной модели контрактования исполнителей;
- выбора адекватной методологии проектирования, разработки и внедрения целевой системы;
- выбора адекватной методологии управления проектом создания целевой системы.

(См. также проблемно-ориентированные уточнения в разделах 6–8).

1.14. **Команда исполнителей**

Риски, связанные с наличием команды исполнителей, адекватной целям, задачам и доступным ресурсам проекта.

1.15. **Согласованность инструментов исполнения и управления проектом**

Риски, связанные с необходимостью обеспечить согласование используемых:

- модели управления проектом;
- технологии проектирования и разработки;
- модели контрактования исполнителей работ по проекту.

1.16. **Процедурная модель учета**

Риски, связанные с выбором процедурной модели учета (в рамках альтернативы *сделочная* или *проводочная* модели учета или же той или иной их комбинации).

1.17. **«Невстраиваемость» ИБ**

Риски «невстраиваемости» штатных систем информационной безопасности в уже имеющиеся ИТ-решения.

1.18. **Баланс «защищенность – надежность – стоимость»**

Риски нарушения баланса «защищенность – надежность – стоимость» формируемого целевого решения.

2. *Архитектура целевых бизнес-процессов* (в том числе модель учета, «сквозные» бизнес-процессы, детальное по-операционное описание бизнес-процессов, формализованные модели данных и процессов и т. д.)

2.1. Неоптимальная **бизнес-архитектура**

Риски выбора (в рамках альтернатив: *централизованная, распределенная, комбинированная*) неоптимальной бизнес-архитектуры в имеющемся конкретном бизнес-контексте:

– зафиксированных базовых характеристик процедур принятия бизнес-решений (в том числе централизованной или распределенной архитектуры риск-менеджмента, налоговой службы и т. п.);
– выбранной топологии распределения бизнес-функций в рамках целевого ландшафта ГО + филиалы (при имеющемся распределении компетенций персонала по этому ландшафту);
– необходимости оптимизировать эксплуатационные расходы, контролировать операционные риски и т. п.

2.2. Неоптимальная **архитектура бизнес-процессов**

Риски выбора неоптимальной архитектуры бизнес-процессов при уже зафиксированном типе бизнес-архитектуры – *централизованной, распределенной, комбинированной*.

2.3. Неадекватная **модель процессов**

Риски выбора (и попыток внедрения) неоптимальной (неадекватной) модели **процессов** (комплекса целевых бизнес-процессов, переход к которым должен стать результатом соответствующего модернизационного проекта).

2.4. **Неполное** описание **целевой бизнес-технологии**

Риски, связанные с неоптимальным выбором и неполнотой описания целевой бизнес-технологии.

2.5. **Инструменты и технология проектирования**

Риски, связанные с проектированием целевых бизнес-процессов (с выбором адекватных инструментов и технологии проектирования).

2.6. **Процессная модель бизнеса**

Риски, связанные с надежностью выбранной процессной модели бизнеса, в том числе с проблемой интеграции процессов при реализации проекта модернизации ИТ-систем банка.

2.7. **Детальность операционных регламентов**

Риски, связанные с недостаточно полным описанием детальных операционных регламентов, необходимых для «настройки» целевой бизнес-технологии средствами выбранного для внедрения прикладного программного обеспечения.

2.8. Особенности **стандартов учета**

Риски, обусловленные различиями в семантике РСБУ[34] и IAS / IFRS[35].

2.9. **Согласованность** трех **уровней интеграции**

Риски, обусловленные недостаточной согласованностью трех уровней интеграции – уровня моделей данных / процессов, уровня

интеграционной шины (ESB)[36], уровня логической структуры хранилища данных (DWH) (см. также проблемно-ориентированные уточнения в разделах 3 и 6).

2.10. **Интеграция разнородных бизнес-процессов**

Риски интеграции разнородных по модели учета бизнес-процессов / цепочек операций (в том числе при организации параллельного учета).

2.11. **Компонентизация**

Риски, связанные с проблемой компонентизации приложений (в том числе с формированием пооперационной структуры бизнес-процессов, необходимой для разработки программных SOA[37]-компонентов).

3. *SW[38]-платформенные технологии*

3.1. Неудачный **выбор ПО**

Риски, связанные с выбором для целевой ИТ-системы неадекватного требованиям прикладного программного обеспечения (формализацией требований, выбором программной платформы, проектированием и внедрением целевой информационной системы, обеспечивающей функционирование соответствующего – см. выше – комплекса целевых бизнес-процессов).

3.2. **Доступность** необходимого **индустриального ПО**

Риски, связанные с наличием на ИТ-рынке адекватного индустриального прикладного программного обеспечения.

3.3. **Функциональная гибкость** ПО

Риски, связанные с недостаточной функциональной гибкостью целевой системы в части используемого прикладного ПО.

3.4. **Масштабируемость** ПО

Риски, связанные с недостаточным уровнем масштабируемости платформенного прикладного ПО (как по функциональности, так и по объемам обрабатываемых данных).

3.5. **Соответствие** ПО **требованиям законодательства РФ**

Риски, связанные необходимостью обеспечить соответствие используемого прикладного ПО требованиям законодательства РФ, в том числе соответствие:

– стандартам РСБУ и требованиям государственных регуляторов;
– требованиям законодательства об авторских правах (так как внесение несанкционированных правообладателем изменений в исходный код разработанного им ПО, вообще говоря, позволяет ему отказаться от предусмотренных законодатель-

ством и практикой рынка обязательств по сопровождению поставленного и измененного ПО);
- требованиям национального законодательства в области обеспечения информационной безопасности, в том числе в части обеспечения банковской тайны и защиты персональных данных.

3.6. **Согласованность уровней интеграции** в части ПО

Риски, связанные с возможностями конкретного платформенного ПО обеспечить согласование трех уровней интеграции (модель данных / процессов, ESB, логическая структура DWH) (см. также проблемно-ориентированные уточнения в разделах 2 и 6).

3.7. Доступность **промышленных ESB-решений**

Риски, обусловленные ограниченным выбором промышленных сертифицированных ESB-решений на российском рынке.

3.8. **Согласованность архитектуры приложений и бизнес-архитектуры**

Риски выбора неоптимальной архитектуры приложений при зафиксированных типе бизнес-архитектуры и архитектуре бизнес-процессов (например, в ходе реализации конкурсных процедур в условиях неполноты требований к целевому решению).

3.9. **Согласованность SW-платформы** и **PM**[39]**-технологии**

Риски несогласованности выбранной SW-платформы для целевого решения с используемой технологией управления проектом (см. также раздел 7).

3.10. **Согласованность SW-платформы и модели контрактования**

Риски несогласованности выбранной SW-платформы для целевого решения с используемой моделью контрактования соисполнителей работ по проекту (см. также раздел 8).

4. *Конфигурация ИТ-ландшафта с точки зрения оптимизации эксплуатационных характеристик и общей стоимости владения*

4.1. Подходящая **системно-техническая архитектура**

Риски выбора неоптимального варианта системно-технической архитектуры при зафиксированных:
- типе бизнес-архитектуры;
- архитектуре бизнес-процессов;
- архитектуре программных приложений

(например, в ходе реализации конкурсных процедур в условиях *неполноты требований* к целевому решению).

4.2. Подходящая **системно-техническая платформа**

Риски, связанные с выбором для целевой ИТ-системы неадекватного варианта системно-технической платформы.

4.3. **Подбор SW-компонентов** для целевого решения

Риски, связанные с подбором необходимых SW-компонентов для полнофункционального целевого решения.

4.4. **Наследование** эксплуатируемого **ПО**

Риски, связанные с наследованием эффективных программных компонентов действующего ИТ-ландшафта в целевую ИТ-систему.

4.5. **Функциональная гибкость системно-технической платформы**

Риски, связанные с недостаточной функциональной гибкостью целевой системы в части системно-технической платформы.

4.6. **Масштабируемость системно-технической платформы**

Риски, связанные с недостаточной масштабируемостью выбираемой системно-технической платформы (в части функциональности или объемов обрабатываемых данных).

4.7. **Соответствие системно-технической платформы** требованиям законодательства РФ

Риски, связанные с необходимостью обеспечить **соответствие** используемой системно-технической платформы требованиям законодательства РФ, в частности требованиям национального законодательства в области обеспечения информационной безопасности, в том числе в части обеспечения банковской тайны и защиты персональных данных.

4.8. **Неприемлемый уровень эксплуатационных расходов**

Риски получить такой уровень бюджетных расходов на эксплуатацию, который окажется неприемлемо высоким (например, в ситуации, когда необходимо постоянно поддерживать целостность программного комплекса, в структуру которого регулярно вносятся те или иные изменения и доработки и т. п.).

4.9. **Неприемлемый уровень операционных рисков**

Риски получить неприемлемо высокий уровень операционных рисков для формируемого целевого решения (например, из-за отказа вендора от сопровождения подвергшегося несанкционированным изменениям базового ПО).

4.10. **Детальность эксплуатационных операционных регламентов**

Риски, связанные с недостаточной проработанностью эксплуатационных регламентов (SLA и т. п.), основная причина которых – отсутствие детальных операционных регламентов для реализуемых бизнес-процессов (в том числе – скрытый характер ряда операционных рисков, присущий недостаточно формализованным – не везде алгоритмически «прозрачным» – бизнес-действиям).

5. *Технологии доработки прикладного SW-решения*

5.1. Легальные **доработки кода**

Риски, связанные с доработками исходного кода базового программного обеспечения (см. также «Соответствие требованиям законодательства РФ» и «Урегулирование взаимоотношений с вендором»)

В том числе:

5.1.1. **Доработки гибкости и масштабируемости ПО**

Риски, связанные с преодолением ограничений в гибкости и масштабируемости целевого решения, обусловленные особенностями платформенного ПО.

5.1.2. **Доработки под законодательство РФ**

Риски, связанные с необходимостью обеспечить (за счет тех или иных доработок) соответствие требованиям законодательства РФ, в том числе:

- стандартам РСБУ и требованиям государственных регуляторов;
- требованиям законодательства об авторских правах (так как внесение несанкционированных правообладателем изменений в исходный код разработанного им ПО, вообще говоря, позволяет ему отказаться от предусмотренных законодательством и практикой рынка обязательств по сопровождению поставленного и измененного ПО) в части технологии взаимодействия с вендором при подобных доработках;
- требованиям национального законодательства в области обеспечения информационной безопасности, в том числе в части обеспечения банковской тайны и защиты персональных данных.

5.1.3. **Легальный статус доработок ПО**

Риски процесса урегулирования взаимоотношений с вендором при доработках исходного кода (см. также «Соответствие требованиям законодательства РФ»).

5.1.4. **Согласование технологий доработок ПО и управления проектом**

Риски несогласованности выбранной технологии доработки целевого решения с используемой технологией управления проектом (см. также раздел 7).

5.1.5. **Согласование технологий доработок ПО и модели контрактования**

Риски несогласованности выбранной технологии доработки целевого решения с используемой моделью контрактования соисполнителей работ по проекту (см. также раздел 8).

5.2. **Стандарты на интерфейсы**

Риски, обусловленные отсутствием на российском рынке банковского ПО общепризнанных стандартов на интерфейсы программных компонентов, формируемых в рамках компонентно-интеграционного подхода.

5.3. **Интеграция справочников**

Риски, сопутствующие интеграции справочников.

5.4. **Сложность EAI-адапторов**

Риски сложности формирования и функциональной негибкости программных адапторов при построении компонентно-интеграционных архитектур EAI[40]-типа.

5.5. **Полнота тестирования ПО**

Риски неполноты (недостаточного уровня) тестирования вновь созданных программных компонентов.

5.6. **Технологичность сборки** целевого решения из **компонентов**

Риски недостаточного уровня технологичности процедур сборки целевого решения из образующих его компонентов, в том числе в части:

- технологичности организации интеграционных тестов;
- возможностей минимизации рисков миграции с текущего решения на целевой ИТ-ландшафт;
- технологичности[41] внесения необходимых изменений в исходный код при реализации тех или иных доработок базового ПО.

5.7. **Баланс изменений** в **исходном коде** и в целевых **бизнес-процессах**

Риски несбалансированности объемов изменений, которые предстоит внести в исходный код (платформенного ПО-решения) и в структуру комплекса целевых бизнес-процессов.

5.8. **Уровень трудозатрат при интеграции разнородных компонентов**

Риски неприемлемого увеличения трудозатрат в ходе организации взаимодействия разнородных программных компонентов при формировании целевого SW-решения (неприемлемая сложность и ресурсоемкость формирования адапторов и т. п.).

5.9. **Интеграция** слабо **компонентизированных приложений**

Риски неприемлемого возрастания трудозатрат и сроков исполнения проектных работ при построении SOA-решения в случае, когда не проведена достаточно детальная компонентизация бизнес-функций целевого решения (и предстоит интегрировать не являющиеся «атомарными» программные компоненты).

6. *Технологии проектирования и разработки*

6.1. **Нештатное прерывание проекта**

Риски нештатного завершения / прерывания проекта внедрения целевой системы (их минимизация), в том числе в части:

- выбора процедур проектирования и разработки, обеспечивающих разделение рисков и ответственность за успех проекта между заказчиком и исполнителями (генеральным подрядчиком и субподрядчиками);
- выбора адекватной методологии проектирования, разработки и внедрения целевой системы.

6.2. Разделение разработки, тестирования и продуктива

Риски, обусловленные недостаточным уровнем надежности контроля ввода создаваемых проектных решений в действующий ИТ-ландшафт банка (например, отсутствие 4-х платформенного[42] системно-технического ландшафта для проекта разработки и внедрения целевого решения).

6.3. **Согласованность** технологий **разработки** и **управления проектом**

Риски несогласованности выбранной модели проектирования и разработки целевого решения с используемой технологией управления проектом (см. также Раздел 7).

6.4. **Согласование уровней интеграции при проектировании и разработке**

Риски проектирования и разработки, связанные с согласованием трех уровней интеграции (модель данных / процессов, ESB, логическая структура DWH) (см. также проблемно-ориентированные уточнения в разделах 2 и 3).

6.5. Критически важная роль **proof of the concept**[43]

Риски разработки (в том числе в соотнесении с рисками внедрения): proof of the concept на системе-прототипе.

6.6. Проблема получения **промежуточных прикладных результатов**

Риски не иметь полезных для бизнес-заказчика промежуточных прикладных (практически значимых) результатов работ по проекту (в том числе бюджетные риски).

6.7. **Компетенции исполнителей**

Риски, связанные с имеющимися в доступе на проекте компетенциями исполнителей, в том числе:

- со структурой *требуемых* и *наличествующих* компетенций[44];
- с *достаточностью* доступных *объемов* требуемых компетенций[45] и др.

6.8. **Неприемлемые сроки и бюджет**

Риски неприемлемого увеличения сроков и бюджетов проекта за счет неоптимального выбора конкретного порядка *перевода разрабатываемых систем / решений* по стадиям проекта (от демо-стадии и стадии проектирования до стадии промышленной эксплуатации).

6.9. **Непрерывность бизнеса** при **миграции** на вновь разработанные решения

Риски нарушения непрерывности бизнеса при реализации процедур миграции с текущего на вновь сформированное (в том числе целевое) решение, а также риски, обусловленные архитектурными особенностями проектного подхода, выбранного для организации проектирования, разработки и внедрения.

6.10. **Неприемлемое количество циклов доработки**

Риски учетных и бухгалтерских служб, связанные с «хроническим» непринятием текущей версии целевого решения в промышленную эксплуатацию ввиду (выявленных в ходе приемосдаточных испытаний) высоких операционных рисков или уровня эксплуатационных расходов.

6.11. **Управляемость формируемой ИТ-системы** в процессно-реальном времени

Риски недостаточного уровня управляемости целевого решения на ландшафте ГО + филиалы из-за отсутствия возможностей реализовать (в рамках приемлемых бюджетов) режим *процессно-реального* времени управления[46].

6.12. **Согласованность** технологии **проектирования / разработки** с моделью контрактования

Риски несогласованности выбранной модели проектирования и разработки целевого решения с используемой технологией контрактования соисполнителей проектных работ (см. также раздел 8).

7. *Технологии управления проектами*

7.1. **Полнота распределения ролей участников проекта**

Риски неопределенности (неназначенности) лиц, исполняющих необходимые роли в структурах управления проектом (в том числе в части принятия решений в случае возникновения конфликта интересов тех или иных сторон).

7.2. **Адекватность** процедуры **конкурсного отбора**

Риски выбора неадекватной процедуры конкурсного отбора исполнителя (и соисполнителей) проекта.

7.3. **Адекватность** технологии **управления проектом**

Риски, связанные со степенью адекватности (решениям, которые приняты по соответствующим вопросам, относимым к разделам 1–6 и 8) выбранной технологии управления проектом.

7.4. **Неуправляемый конфликт интересов** участников проекта

Риски нештатного завершения / прерывания проекта внедрения целевой системы (возможности их минимизации), в том числе в части:

- выбора процедур управления проектом, обеспечивающих разделение рисков и ответственности за успех проекта между заказчиком и исполнителями (генеральным подрядчиком и субподрядчиками);
- выбора адекватной методологии управления проектом создания целевой системы.

7.5. **Взаимодействие структур управления проектом**

Риски, связанные с уровнем надежности контролируемого распространения проектной информации, в том числе между основным проектным офисом и проектным офисом информационной безопасности.

7.6. **Противодействие «полевых командиров»**

Риски, обусловленные противодействием «полевых командиров» (реформируемого банковского бизнеса).

7.7. **Управление взаимоотношениями с вендором** при **доработках исходного кода**

Риски, обусловленные особенностями организации процесса урегулирования взаимоотношений с вендором при доработках исходного кода (см. также «Соответствие требованиям законодательства РФ»).

7.8. **Управление бюджетом и сроками проектных работ**

Риски, связанные с соблюдением согласованных бюджетов и сроков исполнения проекта (в том числе риски, связанные с завышением расценок исполнителями47, и др.).

7.9. **Согласованность PM-технологии с технологией проектирования и разработки**

Риски несогласованности выбранной модели управления проектом с используемой технологией проектирования и разработки целевого решения (см. также раздел 6).

7.10. Проблемы **совместимости стандартной** технологии **проектирования и разработки** с «**экстремальным**» **проектным менеджментом**

Риски «нестыковки» ExtrPM (технологии так называемого экстремального проектного менеджмента, характерного для работы в условиях неполноты требований к целевому решению) со стандартной методологией проектирования и разработки (см. также раздел 6).

7.11. **Согласованность PM-технологии и модели контрактования**

Риски несогласованности выбранной модели управления проектом и задействованной технологии контрактования соисполнителей (см. также раздел 8).

7.12. **Юридически значимая документированность** взаимодействия участников проекта при использовании технологий **прототипирования**

Риски иметь незакрепленными (в юридически значимой форме) промежуточные цели проекта в процессе прототипирования (последовательного масштабирования разрабатываемых систем-прототипов целевого решения).

8. *Технологии контрактования* (в том числе балансировки распределения рисков между заказчиком и исполнителями)

8.1. **Ограниченность стандартной схемы контрактования**

Риски, присущие стандартной схеме контрактования при работе на крупных (комплексных, длительных и высокобюджетных) ИТ-проектах.

8.2. **Контрактные инструменты противодействия досрочному прекращению проекта**

Риск досрочного прекращения проектных работ на той или иной его промежуточной стадии, обусловленный отсутствием надежных инструментов урегулирования конфликтов интересов заказчика и исполнителей, в том числе в части:

- выбора юридически значимой формы разделения рисков и ответственности за успех проекта между заказчиком и исполнителями (генеральным подрядчиком и субподрядчиками);
- выбора адекватной модели контрактования исполнителей.

8.3. **Соответствие модели контрактования** используемой **PM-технологии**

Риски несогласованности выбранной модели контрактования с используемой технологией управления проектом (см. также раздел 7).

8.4. **Соответствие модели контрактования** используемой **технологии проектирования и разработки**

Риски несогласованности выбранных модели контрактования и задействованной технологии проектирования / разработки целевого решения (см. также раздел 6).

8.5. **Баланс рисков между заказчиком и исполнителями**

Угроза неуправляемой группировки важнейших проектных рисков преимущественно на стороне заказчика, в том числе управление балансом *актуальные риски заказчика – стоимость проектных работ* и т. п.

II. Основные группы негативных последствий, возникающих в ситуациях, когда реализуются влияния факторов риска

1. Проект остановился, а его цели не достигнуты.
2. Целевая функциональность не достигнута (реализована лишь частично).
3. Проект не укладывается в рамки запланированных бюджетов.
4. Проект не укладывается в запланированные сроки.
5. Качество полученных решений (функциональных возможностей целевой системы) не соответствует запланированному (заранее согласованному заказчиком и исполнителями) уровню. В том числе:

– в полном объеме целевые требования к разработанной ИТ-системе не достигнуты;
– предлагаемый *de facto* уровень бюджетных расходов на эксплуатацию и сопровождение разработанного решения оказывается неприемлемо высоким;
– фиксируемый *de facto* уровень характерных для разработанного решения операционных рисков оказывается неприемлемо высоким.

III. Примечание

Текущий вариант *Карты Рисков* представляет ее «рабочую» – сокращенную (без потери общности) – версию, из которой удалены очевидные элементы следующего характера.

1. Выделив восемь групп факторов, способных оказать критически важное влияние на результаты любого из проектов рассматриваемого здесь типа, следует (даже без учета эмпирических данных о возможности существования значимых взаимосвязей по крайней мере между некоторыми из них) предположить «ветвление» дальнейших рассмотрений по двум возможным (и, как легко видеть, – взаимоисключающим) направлениям:

– между отдельными риск-факторами (а значит, и любыми их группами) нет каких-либо значимых зависимостей, т. е. риск-факторы взаимно независимы;
– между отдельными риск-факторами (а значит, и некоторыми их группами) возможно существование некоторых значимых зависимостей. То есть на множестве риск-факторов (а вместе с этим – и на множестве групп риск-факторов *Карты Рисков*) могут обнаружиться существенные (для предпринятого нами анализа) зависимости (причем как парные, так и более сложные, «покрывающие» соответствую-

щие риск-факторы сразу из нескольких представленных здесь групп).

2. Далее следовало бы учесть в формируемой *Карте Рисков* оба (только что представленных) варианта «возможных миров»[48]. Однако случай («возможный мир») независимых риск-факторов в плане дальнейшего анализа тривиален (предполагать наличие тех или иных рисков, обусловленных наличием тех или иных связей между риск-факторами, не следует, так как подобных связей, по исходному предположению, просто не существует). То есть его упоминания можно исключить из «рабочего» варианта *Карты Рисков*.

3. В свою очередь равноправный (в плане существования представленных выше «возможных миров») вариант наличия предполагаемых зависимостей следует явным образом представить в «рабочем» варианте *Карты Рисков*, ориентируясь при этом на поиск (выявление и анализ) не только парных, но и более сложных (многофакторых) зависимостей между перечисленными риск-факторами (и их группами).

4. Таким образом, «рабочий» вариант *Карты Рисков* не является предориентированным на поиск соответствующих зависимостей между риск-факторами. Более того, при его выполнении такой анализ не ограничен изучением лишь *попарных* взаимосвязей. Предметом поиска, вообще говоря, здесь должны являться такие (полностью описанные) группы риск-факторов, разумное противодействие влиянию которых обеспечивает приведение соответствующего модернизационного проекта к успеху. Причем в каждую подобную группу взаимосвязанных риск-факторов должны быть включены ***все*** соответствующие (характерные для этой группы) конкретные факторы (адекватное управление которыми ведет проект к успеху) и при этом – только они[49].

Таким образом, предложенный здесь «рабочий» вариант *Карты Рисков* не накладывает каких-либо необоснованных ограничений на структуру «возможных миров» изучаемой нами содержательной предметной области – проблематики результативной организации и управления проектами модернизации ИТ-систем крупных российских коммерческих банков.

Примечания

1 См., например, планы руководства Российской Федерации по развитию национальной экономики на период до 2020 г. (Концепция долгосрочного социально-экономического развития Российской Федерации. М.: Минэкономразвития

РФ, 2008), в том числе по формированию в ближайшей перспективе в Москве международного финансового центра и др.

2 См.: *Забежайло М.И.* Банковский бизнес в России: индустрия или искусство? Волгоград: Волгоградское научное издательство, 2009. 331 с.

3 Разумеется, насколько это возможно.

4 Если это окажется возможным в текущем состоянии детальности описания изучаемой предметной области.

5 Понимая здесь термин «мета-алгоритм» как обобщенное описание семейства схожих между собой алгоритмов, содержащее ряд параметрических величин, точное задание которых формирует ту или иную конкретную версию подобного алгоритма.

6 Имеется в виду методика сбора и анализа данных, принятия решений, проектирования и разработки проектных решений и т. п.

7 Имеется в виду управление проектом, процедуры контрактования исполнителей работ по проекту и т. п.

8 Например: в части сопровождения процессов массового коммерческого кредитования, в части многих типов операций с ценными бумагами и т. п.

9 *Забежайло М.И.* Банковский бизнес в России: индустрия или искусство?

10 См.: *Забежайло М.И.* К вопросу о выборе адекватной методологии трансформации бизнеса крупного коммерческого банка (в печати); *Он же.* О трех уровнях интеграции в современных банковских информационных системах (в печати); *Он же.* О механизмах разделения и оптимизации проектных рисков при модернизации информационных систем крупного коммерческого банка (в печати); *Он же.* О возможностях использования технологии масштабирования прототипов в крупных модернизационных ИТ-проектах (в печати); *Он же.* Программные средства банковских информационных систем: опыт систематизации характеристик представленного на российском ИТ-рынке банковского прикладного программного обеспечения (в печати); *Он же.* Проекты комплексной модернизации в российских банках закончены? (в печати).

11 См.: *Забежайло М.И.* Банковский бизнес в России: индустрия или искусство?

12 Будем далее называть его ЛПР – лицо, принимающее решения.

13 См. вышеуказанные работы М.И. Забежайло.

14 См. вышеуказанные работы М.И. Забежайло.

15 *Забежайло М.И.* Банковский бизнес в России: индустрия или искусство?

16 *Забежайло М.И.* Банковский бизнес в России: индустрия или искусство?; *Он же.* К вопросу о выборе адекватной методологии трансформации бизнеса крупного коммерческого банка; *Он же.* О механизмах разделения и оптимизации проектных рисков при модернизации информационных систем крупного коммерческого банка; *Он же.* О возможностях использования технологии масштабирования прототипов в крупных модернизационных ИТ-проектах.

17 Тривиальными (в предпринятом здесь рассмотрении) представляется естественным считать зависимости уточняющего характера. Пример зависимости подобного типа (см. подробнее описание *Карты Рисков*): фактор 2.4. Неполное

описание целевой бизнес-технологии из раздела 2. Архитектура целевых бизнес-процессов сформированной Карты Рисков представляется естественным рассматривать как проблемно-ориентированное уточнение для фактора 1.3. *Неполнота требований (к целевому решению)* из раздела 1. *Условия общего характера*.

[18] См.: *Забежайло М.И.* Банковский бизнес в России: индустрия или искусство?; *Он же*. К вопросу о выборе адекватной методологии трансформации бизнеса крупного коммерческого банка; *Он же*. О механизмах разделения и оптимизации проектных рисков при модернизации информационных систем крупного коммерческого банка; *Он же*. О возможностях использования технологии масштабирования прототипов в крупных модернизационных ИТ-проектах; *Он же*. Программные средства банковских информационных систем: опыт систематизации характеристик представленного на российском ИТ-рынке банковского прикладного программного обеспечения.

[19] Soft Ware – программные продукты (программное обеспечение).

[20] См.: *Забежайло М.И.* Банковский бизнес в России: индустрия или искусство?; *Он же*. К вопросу о выборе адекватной методологии трансформации бизнеса крупного коммерческого банка; *Он же*. Программные средства банковских информационных систем: опыт систематизации характеристик представленного на российском ИТ-рынке банковского прикладного программного обеспечения; *Он же*. Проекты комплексной модернизации в российских банках закончены?

[21] Service-Oriented Architecture – сервис-ориентированная архитектура программных приложений.

[22] То есть проекты модернизации ИТ-систем в крупных российских коммерческих банках.

[23] Поддерживающего одновременно и стандарты РСБУ, и МСФО. Актуальность подобного рода решений на ближайшую перспективу будет только возрастать. (См., например, принятые руководством РФ решения о переходе на стандарты МСФО к 2015 году. (Д. Медведев подписал закон о переходе на МСФО). [Электронный ресурс] // Сайт РБК: 28 июля 2010 г. [М., 2010]. URL: http://top.rbc.ru/economics/28/07/2010/442240.shtml (дата обращения: 20.12.2010).] и др.).

[24] То есть формирующих (в качестве гипотезы) новое утверждение, логическим следствием которого и оказывается объясняемое исходное утверждение.

[25] См., например, с комбинаторикой влияния выделяемых К. Поппером «***предрасположенностей***, *которые играют роль* ***помех*** *или* ***возмущений***» (*Popper K.R.* A World of Propensities: Two New Views of Causality // World of Propensities. Bristol: Thoemmes, 1990. P. 1–26).

[26] Разумеется, при зафиксированном состоянии множества включенных в рассмотрение фактов (т. е. в конкретной *модели* описания исследуемой предметной области).

[27] В традиционном смысле этого понятия. См.: *Popper K.R.* Conjectures and Refutations: The Growth of Scientific Knowledge. London: Routledge and Kegan Paul, 1963. XIII, 412 p.

[28] См.: *Bernays P.* Concerning Rationality // The Philosophy of Karl Popper (The Library of Living Philosophers. Vol. 14. B. 1 / Ed. by Schlipp P.A.). La Salle, Illionois: Open Count Publishing Co., 1974. P. 597–605; *Popper K.R.* A World of Propensities; *Popper K.R.* Conjectures and Refutations; *Кун Т.* Структура научных революций. М.: АСТ, 2002. 608 с.

[29] См. в том числе, например, применимость Tr^{+}-решения наряду с упоминавшимся выше классическим подходом в ситуациях, когда требования к целевому решению оказываются полностью зафиксированными уже на ранних стадиях реализации соответствующего проекта.

[30] В частности:
- эффективность процессов (Process Efficiency);
- упрощение и стандартизацию процессов (Simplification / Standardization);
- масштабируемость процессов (Scalability);
- интегрированность в бизнес-процессы процедуры риск-менеджмента и внутреннего контроля (Integrated Risk-Management & Compliance);
- минимизированные процессные издержки (Reduced Process Cost).

[31] В частности в части совместимости с новыми версиями базового ПО, сервис-пакетами, релизами и т. п.

[32] Так называемых недекларированных (разработчиком / поставщиком) функциональных возможностей предлагаемого решения.

[33] Один из популярных сегодня примеров оптимизационного решения подобного типа – системно-техническая архитектура вида *Центр Обработки Данных + терминальный доступ*, поддерживающая так называемую *Централизованную Среду Распределенных ИТ-Сервисов* (Central Shared Services Environment).

[34] Российская система бухгалтерского учета.

[35] International Accounting System / International Financial Reporting System – Международная система финансового учета и отчетности.

[36] Enterprise Service Bus – интеграционная шина предприятия.

[37] Service-Oriented Architecture – сервис-ориентированная архитектура программных приложений.

[38] SoftWare – программное обеспечение (ПО).

[39] Project Management – управление проектом.

[40] Enterprise Application Integration – интеграция приложений масштаба предприятия.

[41] В частности в части взаимозависимости и взаимного влияния различных фрагментов исходного кода, существенных при организации доработок базового ПО в соответствии с требованиями государственных регуляторов РФ.

[42] Располагающего пилотной зоной, а также зонами разработки, тестирования и промышленной эксплуатации.

[43] **Proof of the concept** (обоснование концепции) – англоязычный оборот, обычно используемый для обозначения комплекса доказательств состоятельности (правильности, работоспособности, надежности и т. п.) того или иного подхода.

[44] Например, наличие специалистов, владеющих одновременно РСБУ и технологиями IBM BDW при построении КХД (в частности логической структуры КХД, обеспечивающей в том числе подготовку и выпуск обязательной отчетности для ЦБ РФ).

[45] Пример: достаточное количество компетентных специалистов для работы в крупной территориально распределенной структуре (например, на проекте КХД Сбербанка, в структуре которого *Головной Офис + 18 Территориальных Управлений* + ...).

[46] Например, возникновение управленческих ситуаций, когда уже поздно принимать какие-либо решения.

[47] Например, возникновение ценовых завышений в сметах исполнителей, вносимых их риск-менеджментом при работе с «нестандартными» схемами контрактования, отличающимися от традиционно используемых крупными международными ИТ-вендорами.

[48] На необходимость равноправного исследования которых обращал внимание А.С. Есенин-Вольпин при анализе данных в естественно-научных областях знания: *Есенин-Вольпин А.С.* Об антитрадиционной (ультраинтуиционистской) программе оснований математики и естественно-научном мышлении // Семиотика и информатика. Вып. 33. М.: Наука, 1993. С. 13–67.

[49] Эффект своего рода «неподвижной точки»: ***все*** ***те и*** ***только*** ***те***.

П.В. Пекичев

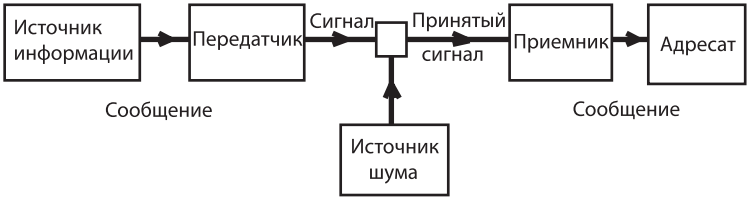
ТЕОРЕТИЧЕСКАЯ ОЦЕНКА ПРОПУСКНОЙ СПОСОБНОСТИ СКРЫТЫХ КАНАЛОВ

Целью статьи является рассмотрение метода оценки количества информации, которая может быть передана в скрытом канале, основанном на вычислении пропускной способности канала передачи данных с шумом. Основное внимание уделяется основной теореме Шеннона для канала с шумом. В данной работе эта теорема доказывается для особого вида шума, который своей деятельностью генерируют обычные пользователи среды передачи данных.

Ключевые слова: скрытый канал передачи данных, теорема Шеннона, пропускная способность, случайный шум, ненадежность.

В 1948 г. вышла статья Клода Элвуда Шеннона «Математическая теория связи»[1], которая положила начало математической теории информации. В этой работе он формализует такие понятия, как «информация», «скорость передачи информации» и «канал передачи данных». Предполагается, что источник данных обладает некоторой статистической структурой (в простейшем случае это дискретный источник, который принимает значение a_i с вероятностью p_i), а скорость создания сообщений отождествляется с мерой неопределенности – энтропией.

Канал передачи данных описывался следующей схемой.



Подробнее рассмотрим каждый элемент схемы.

1. Источник информации, создающий сообщения, которые должны быть доставлены адресату. В данной работе будут рассматриваться только сообщения, состоящие из букв конечного алфавита.

2. Передатчик – устройство, которое перерабатывает входные сообщения в сигналы, соответствующие характеристикам конкретного канала. Это может быть как преобразование звукового сигнала в электрический ток, так и кодирование входного сигнала.

3. Шум – внешнее воздействие, изменяющее передаваемый сигнал.

4. Приемник – устройство, производящее обратные преобразования по отношению к преобразованиям передатчика.

5. Адресат – конечная точка, которой передается сообщение от источника.

Дискретный канал без памяти и с шумом, т. е. канал, у которого передаваемые символы искажаются независимо, задается своими переходными вероятностями $p_j(i)$ – вероятность того, что на выходе обозревается j-й сигнал при условии, что на входе был i-й. Фундаментальным результатом этой статьи является основная теорема Шеннона для каналов с шумом, которая устанавливает связь величины, $C = max_X(H(X) - H_Y(X))$, которая называется пропускной способностью канала, и количества информации, которая может быть передана без ошибок.

Теорема. Пусть дискретный канал обладает пропускной способностью C, а источник энтропией H. Если $H \leq C$, то существует такая система кодирования, что сообщения источника могут быть переданы по каналу со сколь угодно малой частотой ошибок. Если $H > C$, то сообщения источника можно закодировать так, что ненадежность передачи будет меньше, чем $H - C + \varepsilon$, где $\varepsilon > 0$ сколь угодно мало. Не существует способа кодирования, обеспечивающего ненадежность, меньшую $H - C$.

Некоторые обобщения и формализация этой теоремы могут быть найдены в работах[2, 3].

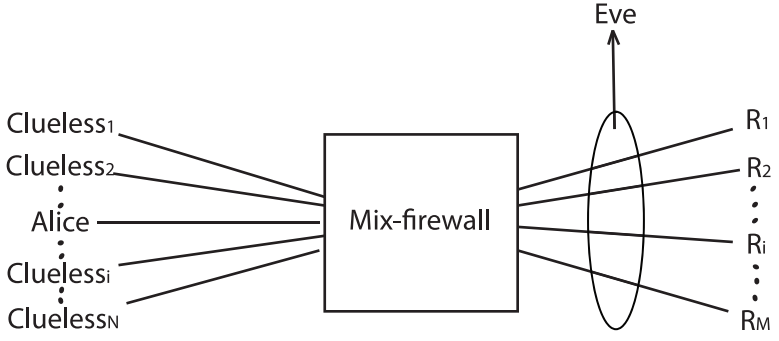
Эта теорема не указывает конкретную безошибочную передачу, но она говорит о существовании такой передачи и об ограничениях на ее скорость.

Однако эта теорема используется не только для анализа качества построенного канала. На ее основе можно сделать вывод и о безопасности информационной системы с точки зрения сохранностей в ней данных. Рассмотрим несколько примеров.

Модели

Целью данной работы является анализ систем передачи данных для выявления возможности построения в них скрытых каналов, т. е. каналов передачи данных, которые не предусматривались разработчиками.

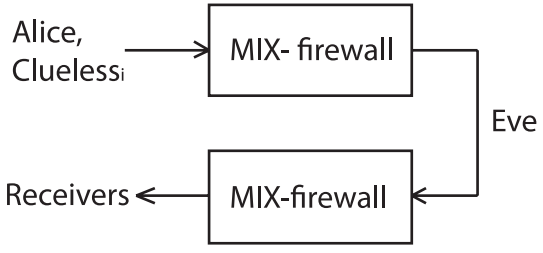
Рассмотрим классическую систему передачи данных. Имеются N «честных» пользователей секретной сети (clueless) и один «злонамеренный» (будем называть его Alice). Пользователи секретной сети имеют доступ к внешним M-серверам (R_i) по уязвимым связям. Данные, находящиеся в сети из n пользователей, не должны быть известны вне этой сети. Пусть имеется дискретное время и в каждый момент времени каждый пользователь посылает (или не посылает) сообщение одному из серверов. Все честные пользователи посылают сообщения в соответствии со своим вероятностным распределением. То есть для каждого пользователя (в том числе и для Alice) задана вероятность $P_i(R_j)$ отправки пользователем i-сообщения на сервер $j = 0...M$ (если $j = 0$, то считаем, что сообщение не было отправлено).



Как видно из рисунка, все сообщения сначала по защищенным связям доставляются на Mix-устройство. Для каждого сообщения микс знает адрес сервера, на который оно должно быть доставлено. И уже из микса закодированные сообщения посылаются по незащищенным связям к серверам. Теперь рассмотрим сообщника Alice, которого будем называть Eve. Eve может просматривать все связи от микса к серверам, но читать закодированные сообщения Eve не может, но каждый момент времени Eve знает количество сообщений, отправленных к каждому серверу (в силу наличия микса Eve не знает, от кого сообщения были отправлены).

Таким образом, имеем канал передачи данных от Alice к Eve, на вход которого передается число от 1 до n, а выходом является целочисленный вектор размерности m. Сообщения, посылаемые честными пользователями, будем считать шумом, искажающим исходное сообщение, передаваемое от Alice. В рассматриваемой системе данный канал является скрытым каналом передачи данных, и оценка его пропускной способности необходима для определения степени защищенности секретной системы из n пользователей от утечек информации из этой системы. Свое же распределение X Alice выбирает так, чтобы максимизировать выражение $H(X) - H_Y(X)$, при этом, в соответствии с основной теоремой Шеннона для каналов передачи данных с шумом, канал передачи от Alice к Eve будет обладать максимальной пропускной способностью. Отметим, что в реальной жизни Alice может представлять из себя программу, которая была установлена на компьютер с секретной информацией, и, действуя, как описано выше, Alice передает эту секретную информацию. Более подробное описание и оценки пропускной способности описанного скрытого канала в некоторых частных случаях можно найти в работах[4, 5].

Аналогичные рассуждения можно провести и для модели с двумя миксами.



В данной модели, благодаря наличию двух миксов, Eve не знает, от кого отправлено сообщение и кому оно отправлено. Единственное, что она может, – это считать количество переданных сообщений за каждый такт времени. Таким образом, вход канала передачи данных от Alice к Eve есть 0 или 1 (что соответствует отправке сообщения от Alice или его отсутствию), а выход – число от 0 до n.

Обобщение

Как видно из описанных примеров, вероятностное поведение честных отправителей описывается дискретной случайной величиной. Но это приближение никак не отражает того, что предпочтения

пользователей к тому или иному серверу могут меняться со временем или случайны отклонения поведения пользователя от заданного распределения.

Докажем несколько важных утверждений. Для этого рассмотрим дискретный канал передачи данных (без памяти) с шумом. Пусть имеются входные символы: 1, ... n и выходные: 1, ... , m. Пусть для каждого момента времени t = 1, 2, ... имеется своя функция $f_t(i,j) = p_t(i|j)$, определяющая вероятности перехода. Таким образом, вместо одной условной энтропии $H_y(X)$ в каждый момент времени будет своя $H_y^t(X)$.

Для описанного канала докажем следующее утверждение.

Утверждение. Рассмотрим канал, у которого существует $\lim_{T\to\infty}(\sum_{t=1}^{T} H_y^t(X)/T) = a(X) \neq 0$. Тогда, если энтропия источника $H(x) \le C = \max_X(H(X) - a(X))$, то сообщения источника могут быть переданы со сколь угодно малой ошибкой. Если же $H(x) > C$, то безошибочная передача невозможна.

Доказательство. Повторяя рассуждения, проводимые при доказательстве основной теоремы Шеннона[6] для канала с шумом, рассмотрим источник, на котором достигается пропускная способность (или становится сколь угодно близкой) C. Рассмотрим всевозможные принимаемые и отправляемые сообщения большой длины T. Из свойств энтропии следует, что отправляемые последовательности распадаются на $2^{TH(X)}$ «высоковероятных» последовательностей и на остальные последовательности, суммарная вероятность которых стремится к 0 при увеличении T. Пусть S – другой источник, создающий информацию со скоростью R, меньшей C. За время T он создаст 2^{TR} высоковероятных сообщений. Каждому из этих сообщений поставим в соответствие высоковероятную последовательность источника S_0 и найдем среднюю вероятность ошибки при передаче в таком классе возможных кодирований. Для каждого выхода канала y мы имеем некоторое число входов канала, из которых могло получиться y (это число будет получено ниже). Если более чем одному такому входу поставлено в соответствие сообщение источника, то имеем ненулевую вероятность ошибки (так как мы не можем точно выбрать между ними). Для усреднения частоты возможных ошибок (здесь под частотой понимается отношение количества выходов, которые могли получиться более чем из одного входа, к общему количеству выходов) по всем кодированиям, каждому такому кодированию припишем одинаковую вероятность. Таким образом, имеем задачу нахождения вероятности ошибки передачи при равномерно распределенных сообщениях

источника S по $2^{TH(X)}$ последовательностям источника S_0. Иными словами, вероятность того, что конкретное сообщение источника S поставлено в соответствие последовательности источника S_0, есть $2^{T(R-H(X))}$. То есть при наблюдении выходной последовательности Y, вероятность того, что никакая другая последовательность не будет сообщением, есть $p(T)=(1-2^{T(R-H(X))})^K$, где K – количество входных последовательностей, из которых могла получиться выходная последовательность Y.

Оценим асимптотический характер поведения K для произвольного Y. Для этого рассмотрим дискретные случайные величины ξ_t: $P(\xi_t=-\log h_t(i|j))=h_t(i,j)$. C их помощью запишем случайную величину, значениями которой являются всевозможные вероятности входных последовательностей длины T, при заданной выходной $\varsigma_T=2^{-\sum_{t=1}^{T}\xi_t}$. Оценим дисперсию случайной величины ξ_t: $D\xi_t=E\xi_t^2-(E\xi_t)^2$. Заметим, что $E\xi_t=H_y^t(x)\le H(x)$.

В силу того, что функция $x\log^2 x$, как несложно показать, является ограниченной на отрезке $[0,1]$ некоторой константой C, то сразу для всех t из интервала $[1,\infty]$ выполняется

$$E\xi_t^2=\sum_{i=1}^{n}\sum_{j=1}^{m}p_t(i,j)\log^2 p_t(i|j)\le\sum_{i=1}^{n}\sum_{j=1}^{m}p_t(i|j)\log^2 p_t(i|j)\le mnC.$$

Так как m и n – фиксированные числа, то $E\xi_t^2$ ограниченно равномерно по t. В силу двух последних замечаний $D\xi_t$ ограничена равномерно по всем t =>. К последовательности $\{\xi_t\}$ применим усиленный закон больших чисел в форме Чебышёва:

$$\lim_{T\to\infty}\left(\frac{\sum_{t=1}^{T}\xi_t}{T}-\frac{\sum_{t=1}^{T}H_y^t(x)}{T}\right)=0$$

с вероятностью 1 (далее все равенства подразумевают равенство на множестве единичной меры)

$$\Rightarrow\lim_{T\to\infty}\frac{\sum_{t=1}^{T}\xi_t}{T}=\lim_{T\to\infty}\frac{\sum_{t=1}^{T}H_y^t(x)}{T}=a(X)\Rightarrow\lim_{T\to\infty}\frac{\sum_{t=1}^{T}\xi_t}{T}=a(X).$$

Следовательно, имеем следующие ограничения на $\sum_{t=1}^{T}\xi_t$: $(a-\varepsilon)T<\sum_{t=1}^{T}\xi_t<(a+\varepsilon)T\Rightarrow 2^{-(a+\varepsilon)T}<\varsigma_T<2^{-(a-\varepsilon)T}$. Так как ς_T выражает возможные вероятности входных последовательностей, то для фиксированного T имеем ограничение на число входных последовательностей, из которых могла получиться фиксированная выходная: $2^{(a-\varepsilon)T}<K<2^{(a+\varepsilon)T}$.

С помощью оценки на K мы можем оценить $P(T)$:

$$(1-2^{-T(R-H(X))})^{2(a(X)+\varepsilon)T} < P(T) < (1-2^{-T(R-H(X))})^{2(a(X)-\varepsilon)T}.$$

Теперь вспомним, что $R < C \Rightarrow R - H(X) = -a(X) - \eta$, где $\eta > 0$

$$(1-2^{-T(a(X)+\eta)})^{2(a(X)+\varepsilon)T} < P(T) < (1-2^{-T(a(X)+\eta)})^{2(a(X)-\varepsilon)T}.$$

Найдем предел левой части неравенства:

$\log(1-2^{-T(a(X)+\eta)})^{2(a(X)+\varepsilon)T} = 2^{(a(X)+\varepsilon)}$

$\log(1-2^{-T(a(X)+\eta)}) = 2^{T(a(X)+\varepsilon)}\,(2^{-T(a(X)+\eta)} + o(2^{-T(a(X)+\eta)})) = 2^{-T(\eta-\varepsilon)} + 2^{T(a(X)+\varepsilon)} o(2^{-T(a(X)+\eta)})$.

В силу того, что η фиксировано, а ε может быть выбрано сколь угодно малым, то предел последнего выражения равен 0, следовательно, предел правой части неравенства равен 1. Аналогично показывается, что предел правой части неравенства также равен $1 \Rightarrow p(T) \to 1$, т. е. средняя вероятность отсутствия ошибки стремиться к 1 =>, средняя вероятность ошибки стремится к 0 => для любого $\varepsilon > 0$ найдется T: средняя вероятность ошибок будет меньше ε. Но если среднее положительных чисел меньше ε, то только та часть из них, доля которых не превышает $\sqrt{\varepsilon}$, может быть больше $\sqrt{\varepsilon}$. То есть при достаточно большом T почти при любом способе кодирования вероятность ошибки будет сколь угодно малой.

Рассмотрим случай, когда $H(X) > C \Rightarrow H(X) = C + b, b > 0 \Rightarrow H(X) > H(X) - H_Y(X) + b \Rightarrow H_Y(X) > b > 0$. То есть каким бы ни было распределение X, неточность передачи будет больше 0, следовательно, точная передача невозможна, что и требовалось доказать.

Теперь рассмотрим случай, когда для каждого момента времени t задана вероятностная мера P_t на множестве функций, $f_{tr}(i,j) = p_{tr}(i|j)$, где $1 \le r \le n_t$, $t = 1, \ldots, \infty$. Этот случай учитывает возможные случайные изменения неопределенности входного сигнала при известном выходном. То есть теперь неточность передачи является случайным процессом: $H_y^t(x)$. В этом случае справедливо следующее утверждение.

Утверждение. Рассмотрим описанный выше канал, у которого существует $\lim_{T\to\infty}(\sum_{t=1}^{T} E(H_y^t(X)/T) = a(X) \neq 0$. Если энтропия источника в секунду $H(x) \leqslant C = \max_X(H(X) - a(X))$, то сообщения источника могут быть переданы со сколь угодно малой неточностью. Если $H(x) > C$, то безошибочная передача невозможна.

Доказательство полностью повторяет предыдущее, за исключением того, что распределение случайной величины ξ_t следует взять таким: $P(\xi_t = -\log p_{tr}(i|j)) = P_t(f_{tr}(i,j))p_{tr}(i|j)$.

Заключение

Доказанные утверждения обобщают понятие неточности передачи с $H_Y(X)$ в случае, когда нет зависимости от времени, до $\lim_{T\to\infty}(\sum_{t=1}^{T} E(H_y^t(X)/T)$ в случае временных и случайных изменений. Это позволит точнее оценить пропускную способность каналов.

Если вспомнить приведенные выше модели, то легко увидеть, что теперь характер поведения реальных людей, обменивающихся данными в описанных сетях, может быть описан не просто случайной величиной, но и случайным процессом. А это дает более точные оценки пропускной способности скрытых каналов, а следовательно, и надежности всей системы в целом.

Примечания

1 См.: *Шеннон К.* Работы по теории информации и кибернетике. М.: Изд-во иностранной литературы, 1963.

2 *Скороход А.В.* Вероятность. Прикладные аспекты // Итоги науки и техники. ВИНИТИ. Современные проблемы математического фундаментального направления. № 43. М.: ВИНИТИ, 1989. С. 188–189.

3 *Добрушин Р.Л.* Общая формулировка основной теоремы Шеннона в теории информации // Успехи математических наук. Т. 14. Вып. 6. 1959. С. 3–104.

4 См.: *Newman Richard E., Nalla Vipan R., Moskowitz Ira S.* Anonymity and Covert Channels in Simple Timed Mix-firewalls. Toronto, Canada: Privacy Enhancing Technologies, 2004. P. 1–16.

5 См.: *Moskowitz Ira S., Newman Richard E., Crepeau Daniel P., Miller Allen R.* Covert channels and anonymizing networks. Washington, DC, USA: WPES, 2003. P. 79–88.

6 См.: *Шеннон К.* Указ. соч.

А.И. Свинцицкий

НЕКОТОРЫЕ СТАТИСТИЧЕСКИЕ СКРЫТЫЕ КАНАЛЫ

Скрытые каналы передачи данных являются существенной угрозой безопасности информации, обрабатываемой в защищенных сегментах сетей, так как практически все используемые в современных сетях передачи данных протоколы имеют уязвимости и являются избыточными. Более того, при их разработке не учитывались вопросы, связанные с возможностью скрытой передачи информации, создания, анализа и противодействия статистическим скрытым каналам. В данной работе рассматривается один из возможных классов статистических скрытых каналов, исследуются его свойства с целью создания эффективных алгоритмов обнаружения и блокировки.

Ключевые слова: скрытые каналы, статистические скрытые каналы, протоколы передачи данных, информационная безопасность.

Пусть имеется N + 1 сегмент локальной сети $S_0, \ldots, S_N$, взаимодействующие друг с другом через сеть Интернет. На пограничных межсетевых экранах или маршрутизаторах применяются механизмы трансляции сетевых адресов (Network Address Translation, NAT), и внешними адресами данных сегментов являются $s_0, \ldots, s_N$ соответственно. При нормальной работе сети возможны задержки при передаче пакетов, которые имеют показательное распределение. На пограничном маршрутизаторе сегмента S_0 установлена программная или программно-аппаратная закладка A, взаимодействующая с сервером нарушителя B, расположеннsм в сети Интернет так, что пакеты из сегмента S_0, адресованные $s_1, \ldots, s_N$, проходят через узел сети, на котором расположен этот сервер. Время на закладке A и сервере B синхронизировано с использованием протокола NTP (Network Time Protocol). Цель – передать информацию (хотя бы один бит) от закладки A серверу B.

Алгоритм передачи одного бита информации

Обобщим алгоритм, предложенный в работе А.А. Грушо и Е.Е. Тимониной «Статистические скрытые каналы»[1], и исследуем его свойства.

А и В имеют некоторый секретный ключ, который состоит из следующих величин:

n – длина эксперимента;

$\{t_1, \dots, t_n\}$, где $t_1 < t_2 < \dots < t_n$ – время эксперимента;

$\delta > 0$ – интервал единичного эксперимента;

$C > 0$ – константа.

Закладка А формирует специальную последовательность исходящих пакетов, адресованных $s_1, \dots, s_N$. Сервер В отслеживает, какие пакеты проходят через него в каждый момент времени $(t_i - \delta, t_i + \delta)$, $i = \overline{1, n}$, и вычисляет следующие величины:

$$v(S_{i1}) = \sum_{i=1}^{n} I\{s_{i1} \in (t_i - \delta, t_i + \delta)\};$$

$$v(S_{i2}) = \sum_{i=1}^{n} I\{s_{i2} \in (t_i - \delta, t_i + \delta)\};$$

Если $v(S_{i1}) < C \le v(S_{i2})$, то сервер В делает предположение, что закладка А передает «1».

Если $v(S_{i2}) < C \le v(S_{i1})$, то делается предположение, что передается «0».

Для того чтобы сервер В мог отделить наличие передачи данным от нормальной работы сети, необходимо оценить длину эксперимента n.

Составим две гипотезы:

H_0: нормальная работа сети, передачи нет ($v < C$);

H_1: идет передача информации ($v \geqslant C$);

α – уровень значимости;

β – вероятность ошибки второго рода.

Оценим константу С исходя из предположения, что вероятность ошибки первого рода (H_0 неверно отвергнута) не должна быть выше, чем α, т. е.:

$$P_0 \left\{ \sum_{i=1}^{n} I\{s \in [t_i, t_i + \delta)\} \geqslant C_\alpha \middle| \textit{Сигнала нет} \right\} < \alpha.$$

Так как при нормальной работе все пакеты в сети появляются независимо и равномерно, $P_0\{s \in [t_i, t_i + \delta)\} = P_0\{s \in [0, +\delta)\} = p_0$, то

$$p_0 = \frac{1}{N}\,\delta, \text{ откуда}$$

$$P_0\left\{\sum_{i=1}^{n} I\{s\in[0,\delta)\}\geqslant C_\alpha\right\}<\alpha$$

$$P_0\left\{\frac{\sum_{i=1}^{n} I\{s\in[0,\delta)\}-E_0 I\{s_i\in[0,\delta)\}}{\sqrt{D_0\sum_{i=1}^{n} I\{s\in[0,\delta)\}}}\geqslant\frac{C_\alpha-E_0\sum_{i=1}^{n} I\{s\in[0,\delta)\}}{\sqrt{D_0\sum_{i=1}^{n} I\{s\in[0,\delta)\}}}\right\}<\alpha.$$

Применив центральную предельную теорему, получим:

$$1-\Phi\left(\frac{C_\alpha-E_0\sum_{i=1}^{n} I\{s\in[0,\delta)\}}{\sqrt{D_0\sum_{i=1}^{n} I\{s\in[0,\delta)\}}}\right)<\alpha;$$

$$\Phi\left(\frac{C_\alpha-E_0\sum_{i=1}^{n} I\{s\in[0,\delta)\}}{\sqrt{D_0\sum_{i=1}^{n} I\{s\in[0,\delta)\}}}\right)<1-\alpha;$$

$$\frac{C_\alpha-E_0\sum_{i=1}^{n} I\{s\in[0,\delta)\}}{\sqrt{D_0\sum_{i=1}^{n} I\{s\in[0,\delta)\}}}>\Phi^{-1}(1-\alpha).$$

Обозначим $X_\alpha=\dfrac{C_\alpha-E_0\sum_{i=1}^{n} I\{s\in[0,\delta)\}}{\sqrt{D_0\sum_{i=1}^{n} I\{s\in[0,\delta)\}}}$, откуда

$$X_\alpha>\Phi^{-1}(1-\alpha)$$

$$C_\alpha>\Phi^{-1}(1-\alpha)\times\sqrt{D_0\sum_{i=1}^{n} I\{s\in[0,\delta)\}}+E_0\sum_{i=1}^{n} I\{s\in[0,\delta)\}.$$

Из полученной оценки для C_α получим оценку для n из предположения, что вероятность ошибки второго рода не должна быть выше β:

$$P_1\left\{\sum_{i=1}^{n} I\{s\in[t_i,t_i+\delta)\} < C_\alpha \,|\, Сигнал\ есть\right\} < \beta.$$

Так как единственное ограничение при передаче пакетов через сеть – это задержка, то можно переписать данное неравенство, опираясь на $P_1\{s\in[t_i,t_i+\delta)\} = P_1\{s\in[0,\delta)\} = p_1$ и на то, что задержка распределена показательно: $p_1 = 1 - e^{-\lambda\delta}$.

$$P_1\left\{\sum_{i=1}^{n} \mathrm{I}\{s\in[0,\delta)\} < C_\alpha\right\} < \beta$$

$$P_1\left\{\frac{\sum_{i=1}^{n} I\{s\in[0,\delta)\} - E_1\sum_{i=1}^{n} I\{s\in[0,\delta)\}}{\sqrt{D_1\sum_{i=1}^{n} I\{s\in[0,\delta)\}}} < \frac{C_\alpha - E_1\sum_{i=1}^{n} I\{s\in[0,\delta)\}}{\sqrt{D_1\sum_{i=1}^{n} I\{s\in[0,\delta)\}}}\right\} < \beta$$

$$\frac{C_\alpha - E_1\sum_{i=1}^{n} I\{s\in[0,\delta)\}}{\sqrt{D_1\sum_{i=1}^{n} I\{s\in[0,\delta)\}}} < \Phi^{-1}(\beta)$$

Подставим $C_\alpha = X_\alpha \times \sqrt{D_0\sum_{i=1}^{n} I\{s\in[0,\delta)\}} + E_0\sum_{i=1}^{n} I\{s\in[0,\delta)\}$

$$E_0\sum_{i=1}^{n} I\{s\in[0,\delta)\} = np_0$$

$$E_1\sum_{i=1}^{n} I\{s\in[0,\delta)\} = np_1$$

$$D_0\sum_{i=1}^{n} I\{s\in[0,\delta)\} = np_0(1-p_0)$$

$$D_1\sum_{i=1}^{n} I\{s\in[0,\delta)\} = np_1(1-p_1)$$

$$p_1 > p_0$$

Получаем $\frac{X_\alpha \times \sqrt{np_0(1-p_0)} + np_0 - np_1}{\sqrt{np_1(1-p_1)}} < \Phi^{-1}(\beta)$

$$X_\alpha < \frac{\Phi^{-1}(\beta)\times\sqrt{np_0(1-p_0)} - np_0 + np_1}{\sqrt{np_0(1-p_0)}}$$

$$\Phi^{-1}(1-\alpha) < \mathrm{X}_\alpha$$

$$\Phi^{-1}(1-\alpha) < \frac{\Phi^{-1}(\beta)\times\sqrt{np_0(1-p_0)} - np_0 + np_1}{\sqrt{np_0(1-p_0)}}$$

$$\Phi^{-1}(1-\alpha) < \frac{\Phi^{-1}(\beta)\times\sqrt{p_0(1-p_0)} - \sqrt{n}p_0 + \sqrt{n}p_1}{\sqrt{p_0(1-p_0)}}$$

$$\sqrt{n}(p_0 - p_1) > \Phi^{-1}(1-\alpha)\times\sqrt{p_0(1-p_0)} - \Phi^{-1}(\beta)\times\sqrt{p_1(1-p_1)},$$

откуда получаем, что $n > \left(\frac{\Phi^{-1}(1-\alpha)\times\sqrt{p_0(1-p_0)} - \Phi^{-1}(\beta)\times\sqrt{p_1(1-p_1)}}{p_1 - p_0}\right)^2$

Алгоритм передачи k бит информации.

Расширим предложенный алгоритм для возможности передачи за один эксперимент длины *n k* бит информации.

Пусть в каждый момент времени сервер *B* наблюдает за пакетами, адресованными получателям $S_{i_1}, \ldots, S_{i_k}$, и вычисляет величины

$$v(S_{i_j}) = \sum_{i=1}^{n} I\{s_{i_j} \in (t_i - \delta, t_i + \delta)\}, j = \overline{1, k}.$$

Все множество $\{v(S_{i_j})\}$ разбивается на два непересекающихся подмножества

$$C_1 = \{v(S_{i_j}) : v(S_{i_j}) < C\}, j = \overline{1, k};$$

$$C_2 = \{v(S_{i_j}) : v(S_{i_j}) \geqslant C\}, j = \overline{1, k}.$$

Всего существует C_k^p вариантов разбиения на множества мощности p и $k-p$, существует $\sum_{i=0}^{k} C_k^i = 2^k$ вариантов разбиения на два непересекающихся множества, что соответствует мощности E_2^k. Можно построить взаимно-однозначное соответствие между всевозможными разбиениями на множества C_1, C_2 и векторами длины k.

Для полного описания работы предложенного канала рассмотрим алгоритм старта, обучения, изменения ключа и остановки передачи информации.

1. Алгоритм старта: хакладка А точно знает, что пакет адресованные S_1 анализируются на сервере В. Создается специальная последовательность пакетов, такая, что $v(S_1) \geqslant C$ в первых K экспериментах и $v(S_1) < C$ в следующем эксперименте. Вероятность наступления такого события при нормальной работе сети будет меньше $\frac{\alpha}{K}$. Таким образом, со следующего эксперимента можно начинать передачу данных.

2. Первоначальный набор адресов $S_{i_j}, \ldots, S_{i_k}$, используемых для передачи информации, формируется на основании ключа.

3. После каждого эксперимента происходит изменение адресов, на которых передается информация по следующему алгоритму:

a. Закладка A и сервер B знают некоторый автомат (X, K, Q), где $X=\{S_1,\dots,S_N\}$, $M=\{0,1\}$, значение выбирается исходя из условия 1 – количество 1 в передаваемом векторе $\{\lambda_1,\dots,\lambda_N\}$ больше количества 0, 0 – иначе.

$$Q: \underbrace{X\times\dots\times X}_{k}\times M\to \underbrace{X\times\dots\times X}_{k}.$$

b. После каждого эксперимента на A и B вычисляется новый набор адресов

$$(S'_{i_1},\dots,S'_{i_k})=Q(S'_{i_1},\dots,S'_{i_k},m),\ m\in M.$$

c. Так как рассматриваемый канал является каналом с недостоверной доставкой (Z-канал), то необходимо периодически синхронизировать состояния автоматов. Можно реализовать это путем передачи каждые K экспериментов синхронизирующей последовательности, т. е. каждые K экспериментов мы передаем номер состояния в котором находится автомат на закладке A.

4. При окончании передачи посылается последовательность, идентичная старту.

В статье рассматривается алгоритм статистического скрытого Z-канала передачи данных, который можно создать, используя знания о сетевых протоколах и «нормальной» работе сети передачи данных. Были получены оценки длины эксперимента, позволяющего различить две гипотезы с заданными уровнями ошибок первого и второго родов. Исследуя вопросы работы сети, можно создавать эффективные алгоритмы обнаружения и предотвращения статистических скрытых каналов.

Примечания

[1] *Грушо А.А., Тимонина Е.Е.* Статистические скрытые каналы // Материалы XVII Общероссийской научной конференции «Методы и технологические средства обеспечения безопасности информации» 7–11 июля 2008 г. СПб.: Изд-во Санкт-Петербургского политехнического университета, 2008. С. 44–45.

А.А. Грушо, Н.А. Грушо, Е.Е. Тимонина

ИСКУССТВЕННАЯ НЕДОСТОВЕРНОСТЬ ИНФОРМАЦИИ КАК СРЕДСТВО ЕЕ ЗАЩИТЫ

В работе рассмотрены основные принципы использования недостоверных данных для организации защиты информации. Показано, что с помощью недостоверности можно решать задачи защиты конфиденциальности, целостности и доступности. Рассмотрены основные методы внесения недостоверности и оценки эффективности методов использования недостоверной информации.

Работа выполнена при поддержке РФФИ, грант № 10-01-00480.

Ключевые слова: защита информации, недостоверность информации, интеллектуальный шум.

Введение

Данная работа является продолжением нашей публикации[1].

Информация циркулирует по стандартному циклу: сбор информации – анализ информации – принятие решений по результатам анализа – внедрение решений

Ценность информации определяется возможностью ее использования для анализа и принятия решений. Достоверность информации и правильный анализ обеспечивают, как правило, правильное решение, т. е. возможность с помощью правильного решения добиться заданной цели. Если информация недостоверна, то уверенности в правильных результатах анализа нет и основанное на результатах анализа решение может оказаться неправильным, т. е. не приведет к поставленной цели.

Модель недостоверности информации для защиты конфиденциальности

Рассмотрим применение принципа недостоверности информации для защиты конфиденциальности, целостности и доступности.

Пусть X – множество решений, P_0 – распределение вероятностей выбора решений по результатам анализа достоверной информации, P_1 – распределение вероятностей выбора решений на основе анализа ложной информации. Тогда распределение вероятностей

$$\lambda P_0 + (1-\lambda) P_1,$$

где $0 \leqslant \lambda \leqslant 1$, соответствует выбору решения на основе анализа недостоверной или неполной информации. То есть недостоверность информации – это отсутствие уверенности в ее правильности.

Если $P(X)$ – распределение вероятностей на множестве решений без какой бы то ни было исходной информации, а $P(X|Y)$ – распределение вероятностей на множестве решений при условии полученной информации Y, то идеальная конфиденциальность по Шеннону[2] определяется равенством

$$P(X) = P(X|Y).$$

Это равенство выполняется в том случае, когда используется совершенный шифр, не позволяющий аналитику получить истинную информацию для принятия решения.

Можно по-другому подойти к усложнению работы аналитика, не используя криптографию. В случае когда информация достоверна, распределение $P(Y)$ является вырожденным, т. е. принимает значение 1 на заданной достоверной информации Y. В общем случае заведомо

$$P(X) \neq P(X|Y).$$

Однако по формуле полной вероятности

$$P(X) = P(Y) \cdot P(X|Y).$$

В случае недостоверности исходной информации распределение $P(Y)$ не является вырожденным и в идеальном случае может породить такой же эффект неопределенности в выборе решения, как при использовании криптографии.

Таким образом, искусственная недостоверность информации может защищать конфиденциальность достоверных данных.

Модель недостоверности информации для защиты целостности

Пусть легальная информация встроена в некоторый легальный контейнер с помощью метода стеганографии[3]. На приемном конце стеговставка выделяется из контейнера, а контейнер отбрасывается. Стеговставка обрабатывается по алгоритму для входящих сообщений и передается пользователю. Пусть используемый стеганографический метод зависит от ключа, известного на передающем и приемном концах. Тогда выделенная стеговставка, содержащая легальную информацию, располагается в контейнере так, что противник не может ее выделить.

Это значит, что противник не имеет достоверной информации о местоположении истинной информации в сообщении. Нарушение целостности, как правило, связано с внесением искажений в передаваемую информацию. Так как противник не знает, где расположена истинная информация, он вынужден вносить искажения случайно, а значит, повредит контейнер. Искажения контейнера позволяют выявлять возможные нарушения целостности истинного сообщения и затребовать ее повторной передачи. Даже если противник повторно искажает новую передачу, он с большой вероятностью искажает другие знаки истинного сообщения. Тогда при третьей передаче, на которую также воздействует противник, методом голосования истинное сообщение восстанавливается.

Модель недостоверности информации для защиты доступности

Пусть аналитику передается несколько сообщений, из которых он по ключу может выбрать истинное сообщение. Если при этом сообщения передаются по разным каналам, то противнику для того, чтобы обеспечить недоступность, нужно перекрыть все каналы, что является значительно более сложной задачей. При этом выделить истинную информацию противник не может даже в том случае, когда для защиты данных не используется криптография.

Методы внесения недостоверности в информацию

Наиболее эффективно вносить недостоверность на этапе сбора информации. Любой информационный объект можно представить в виде набора переменных. Конкретная информация представляет

собой набор значений этих переменных. Сбор информации представляет собой считывание значений этих переменных. Отсюда вытекают следующие способы внесения недостоверности на этапе сбора данных:

- искажение или изменение значений переменных;
- ликвидация переменных (отсутствие соответствующих значений в передаваемых данных);
- создание ложной переменной (появление значений ложной переменной чаще всего создает противоречивость исходных данных).

Например, для идентификации работы враждебного кода в компьютерной системе программа защиты должна идентифицировать несколько событий. Если какие-либо события отсутствуют, то достоверного решения о наличии враждебного кода программа принять не может.

Иногда создание недостоверной информации может представлять собой сложную задачу. В частности, это требуется, когда ложный информационный объект должен выглядеть вполне правдоподобно, т. е. удовлетворять ряду логических ограничений, связанных с его достоверностью. Например, тестирование программ на истинность исходных данных запрещено, следовательно, необходимо создание правдоподобных данных для тестирования, которые являются заведомо недостоверными.

Наибольшую сложность в создании недостоверной информации представляют процессы накопления информации. В процессе накопления, как правило, увеличивается количество логических ограничений на данные, что увеличивает вероятность выявления недостоверности информации. Однако эта проблема решается с помощью создания моделей информационных объектов (например, UML-моделей[4]).

Оценка эффективности методов защиты информации с помощью внесения недостоверности

Оценка эффективности защиты должна строиться исходя из целей внесения недостоверности.

Мы предлагаем следующие уровни оценки эффективности:

- ложное решение является более предпочтительным;
- одинаковые доводы «за» и «против» правильного и ложного решения;
- невозможность доказательства недостоверности информации, используемой для решения.

Заключение

Целью данной работы являлось привлечение внимания к нетрадиционным методам и средствам защиты информации, таким как внесение недостоверности. Предоставление противнику недостоверной информации широко используется в военных целях, однако эти методы также могут эффективно применяться в конкурентной борьбе и при решении политических задач.

Примечания

[1] См.: *Грушо А.А., Тимонина Е.Е.* Интеллектуальный шум // Проблемы информационной безопасности. Компьютерные системы. № 1. СПб.: Изд-во Санкт-Петербургского политехнического университета, 2000.

[2] См.: *Шеннон К.* Работы по теории информации и кибернетике. М.: Иностранная литература, 1963. 829 с.

[3] См.: *Грушо А.А., Грушо Н.А., Тимонина Е.Е.* Некоторые применения стеганографии и защищенность стегосхем // Проблемы информационной безопасности. Компьютерные системы. № 2. СПб.: Изд-во Санкт-Петербургского политехнического университета, 2007.

[4] См.: *Буч Г., Якобсон А., Рамбо Дж.* UML. Классика CS. 2-е изд. / Пер. с англ. Под общ. ред. С. Орлова. СПб.: Питер, 2006. 736 с.

С.М. Иглицкая

К ВОПРОСУ СТРУКТУРНО-АЛГЕБРАИЧЕСКОГО И СЕМАНТИКО-ПРАГМАТИЧЕСКОГО АНАЛИЗА МУЗЫКАЛЬНОГО ТЕКСТА

Статья посвящена некоторым аспектам исследования музыкального текста в плане оценки эффективности музыкального и вербального каналов коммуникации. Основные разделы посвящены сравнительному анализу нулевого и первого приближений модели дискретного сообщения Дж. фон Неймана вербального текста и музыкального текста так называемого строгого стиля. В завершение затрагиваются некоторые вопросы семантической содержательности музыкальных произведений.

Ключевые слова: текст музыкальный, текст вербальный, канал коммуникации, модель сообщения Дж. фон Неймана, конечный алфавит, семантика текста.

Введение

Спектр проблем, связанных с анализом и моделированием музыкального текста (МТ), представляет собой обширную и малоизученную область. Обзор искусствоведческих источников показывает, что существующая методология и инструментарий не позволяют решить целый круг задач, касающийся естественнонаучной стороны изучения МТ. Исследование же его с позиций точных наук, напротив, способно пролить свет, в том числе, и на некоторые чисто гуманитарные проблемы.

В рамках данной статьи приведен один из возможных подходов к подобному исследованию: МТ рассматривается в сравнении с вербальным текстом (ВТ). Целью работы является сравнительный анализ музыкального и вербального каналов коммуникации как в структурно-алгебраическом, так и в семантико-прагматическом аспекте в соответствии с задачами разработки и исследования мето-

дов и алгоритмов анализа текста и создания языков представления знаний для плохо структурированных предметных областей[1], а также для исследования возможности использования МТ для создания скрытого канала вербальной коммуникации[2].

Принципиальное отличие МТ от ВТ состоит в его изначальной художественной направленности: если ВТ может выполнять как художественно-эстетические, так и информационно-содержательные функции, то МТ по своей природе предназначен в первую очередь для художественного воздействия, информационная же составляющая МТ исследована весьма поверхностно, используется крайне редко и в основном в опосредствованном виде. Кроме того, структура МТ, по сравнению с ВТ, гораздо более сложна – в первую очередь вследствие ее нелинейности.

Так как любой канал передачи информации обладает ограниченной пропускной способностью и, таким образом, количество информации, передаваемой в единицу времени, всегда лимитировано, то встает вопрос об эффективности того или иного способа записи данных, обеспечивающего передачу максимального их количества за минимальное время. Причем речь идет не только о механистической задаче повышения пропускной способности канала связи, но и о целом ряде прагматических задач, связанных с разрешением проблемы перехода количества в качество, когда заданный объем структурированной информации в единицу времени формирует вполне определенную эмоциональную реакцию субъекта (интеллектуальной системы).

В связи с этим возникает сопутствующая проблема, связанная со способом записи соответствующих текстов. Если для ВТ он достаточно ясен – имеется конечный алфавит, состоящий из букв, цифр, специальных символов и т. д., и за каждым отрезком текста может следовать с некоторой вероятностью (или закономерностью) любая из букв этого алфавита, то в МТ каждая очередная нота может иметь любую высоту, любую длительность, тембр, громкость (причем это не только в абстрактной обобщенной модели – подобный подход нередко встречается в композиторских техниках XX в.).

Следовательно, необходимо каким-либо образом упростить МТ до уровня модели конечного алфавита и только после этого проводить его сравнительный анализ с ВТ с точки зрения количества информации.

I. Основной терминологический аппарат

Приведем некоторые определения и формальные соотношения, используемые настоящей работе[3].

Канал передачи данных – средства двухстороннего обмена данными, которые включают в себя линии связи и аппаратуру передачи (приема) данных. Каналы передачи данных связывают между собой источники информации и приемники информации. Будем рассматривать только идеальные каналы, где между элементами кодовых сигналов на входе и выходе существует однозначное соответствие (ошибки в канале отсутствуют).

Сообщением называют информацию, выраженную в определенной форме и предназначенную для передачи от источника к адресату. *Дискретное сообщение* есть конечная последовательность отдельных символов. Формирование дискретных сообщений рассматривают как последовательный случайный выбор того или иного символа из алфавита источника сообщений, т. е. как формирование дискретной случайной последовательности.

Понятие количества информации по К. Шеннону определяется следующим образом. Пусть мощность алфавита A источника сообщений $|A| = m$. Если вероятности появления всех символов одинаковы и равны $P = 1/m$, то количество информации, которое переносит символ, выражается формализмом $I_1 = \log m = -\log P$.

При появлении символов a_i с различными вероятностями $P(a_i)$ $I_i = -\log P(a_i)$.

Энтропией источника дискретных сообщений называется среднее количество информации $H(A)$, которое приходится на один символ источника. Для ее вычисления используется формула Шеннона:

$$H(A) = -\sum_{i=1}^{m} P(a_i) \log P(a_i).$$

Если символы являются взаимосвязанными (коррелированными друг с другом), используется понятие *условной энтропии*:

$$H(A'/A) = -\sum_{i=1}^{m} P(a_i) \sum_{j=1}^{m} P(a'_j/a_i) \log P(a'_j/a_i),$$

где $P(a'_j/a_i)$ – условная вероятность появления символа a_j после символа a_i.

Из-за корреляционных связей символов и неравновероятного их появления в реальных сообщениях снижается среднее количество информации, которое переносит один символ.

Эти потери информации характеризуются *коэффициентом избыточности*

$$r = (H_1 - H) / H / \log m,$$

где H_i – максимальное количество информации, которое может переносить один символ, H – количество информации, которое переносит один символ в реальных сообщениях.

Наиболее часто основание логарифма в приведенных формулах принимают равным 2. При этом единицей количества информации является *бит* (*binary digit*).

Производительностью источника сообщений называется среднее количество информации, выдаваемой источником в единицу времени, а именно $H' = H/t$ [бит/с].

Для каналов передачи информации вводят аналогичную характеристику – скорость передачи информации C. Максимальное ее значение называется пропускной способностью канала $C = V_{\max} H$, где V – скорость передачи электрических кодовых сигналов, H – энтропия сообщения.

Обозначим через C_v пропускную способность канала вербальной коммуникации, а C_m – пропускную способность канала музыкальной коммуникации.

Выдвигается *следующая конструктивная гипотеза*: $C_m > C_v$.

II. Модель фон Неймана различных приближений и ее применение к вербальному тексту

Пусть имеется некоторый конечный алфавит A мощности $|A| = Z$. Последовательная семиотическая модель Дж. фон Неймана[4] представляет собой конечные (длины l) последовательности символов алфавита A:

$$A : \sigma_i = \sigma_{i_1}, \ldots, \sigma_{i_l}, l \leqslant L \leqslant \infty, \sigma_{i_j} \in A, |A| = z, z \leqslant Z \leqslant \infty$$

для $\forall i, j, j = \overline{1,l}, i = \overline{1,N}, N \leqslant \sum_{k=1}^{L} Z^k$.

Нас будет интересовать количество всех порождаемых алфавитом A слов, длина которых не превосходит заданного l.

В *модели нулевого приближения* не вводится никаких ограничений на порядок следования символов алфавита A – за любым символом может следовать любой. Тогда, как нетрудно видеть, количество слов длины, не превышающей l, выражается следующим образом:

$$S(l) = \sum_{k=1}^{L} Z^k. \qquad (1)$$

В *модели первого приближения* имеются ограничения на пары подряд идущих символов (запретные биграммы), определяемые бинарной матрицей размерности $Z \times Z$ $B = (\{b_{ij}\})$, $i, j = 1, Z$, где $b_{ij} = 1$ для разрешенной и $b_{ij} = 0$ – для запрещенной пары символов алфавита A, $|A| = Z$.

Для вычисления количества слов длины не больше l в *модели первого приближения* используется следующая формула[5]:

$$S(l) = \sum_{k=2}^{L} \left(\sum_{i,j=1}^{Z} b_{ij}^{(k-1)} \right) + Z. \qquad (2)$$

В общем случае в *модели n-го приближения* вводятся ограничения на использование n-1 последовательных символов.

Рассмотрим применение модели фон Неймана нулевого и первого приближений к русскому языку. Вычисление количества слов в модели нулевого приближения не составляет труда. Для вычисления количества слов в модели первого приближения использована разработанная программа, создающая на основе доступных словарей русского языка[6] матрицу запретных биграмм 33×33 и вычисляющая количество слов в модели первого приближения по формуле (2) (табл. 1).

Таблица 1

Количество слов в модели фон Неймана
нулевого и первого приближений для русского языка

Длина (I)	$S_0(I)$	$S_1(I)$	$\lg(S_0(I)/S_1(I))$
1	33	33	0,00
2	1089	866	0,10
3	37026	24392	0,18
4	1222947	662427	0,27
5	40358340	17967831	0,35
6	1331826309	487341660	0,44
7	43950269286	13218129133	0,52
8	$1{,}45036 \cdot 10^{12}$	$3{,}58514 \cdot 10^{11}$	0,61
9	$4{,}78618 \cdot 10^{13}$	$9{,}72396 \cdot 10^{12}$	0,69
10	$1{,}57944 \cdot 10^{15}$	$1{,}63742 \cdot 10^{14}$	0,78
11	$5{,}21215 \cdot 10^{16}$	$7{,}15346 \cdot 10^{15}$	0,86
12	$1{,}72001 \cdot 10^{18}$	$1{,}94023 \cdot 10^{17}$	0,95
13	$5{,}67604 \cdot 10^{19}$	$5{,}26246 \cdot 10^{18}$	1,03

Окончание таблицы 1

Длина (I)	$S_0(I)$	$S_1(I)$	$\lg(S_0(I)/S_1(I))$
14	$1{,}87309 \cdot 10^{21}$	$1{,}42733 \cdot 10^{20}$	1,12
15	$6{,}1812 \cdot 10^{22}$	$3{,}87135 \cdot 10^{21}$	1,20
16	$2{,}0398 \cdot 10^{24}$	$1{,}05002 \cdot 10^{23}$	1,29
17	$6{,}73133 \cdot 10^{25}$	$2{,}84797 \cdot 10^{24}$	1,37
18	$2{,}22134 \cdot 10^{27}$	$7{,}72453 \cdot 10^{25}$	1,46
19	$7{,}33042 \cdot 10^{28}$	$2{,}09512 \cdot 10^{27}$	1,54
20	$2{,}41904 \cdot 10^{30}$	$5{,}68257 \cdot 10^{28}$	1,63
21	$7{,}98283 \cdot 10^{31}$	$1{,}54128 \cdot 10^{30}$	1,71
22	$2{,}63433 \cdot 10^{33}$	$4{,}1804 \cdot 10^{31}$	1,80
23	$8{,}6933 \cdot 10^{34}$	$1{,}13385 \cdot 10^{33}$	1,88
24	$2{,}86879 \cdot 10^{36}$	$3{,}07532 \cdot 10^{34}$	1,97
25	$9{,}467 \cdot 10^{37}$	$8{,}34117 \cdot 10^{35}$	2,05

В таблице $Si(l)$ – количество слов длины не больше l в модели i-го приближения; последний столбец есть логарифмическая шкала отношения количества слов в модели нулевого приближения к количеству слов в модели первого приближения.

Из табл. 1 видно, что количество запретных пар символов для русского языка не столь велико, и, вследствие этого, даже на длине 25 разница порядков (в логарифмической шкале) для количества слов длины не больше данной в моделях нулевого и первого приближений равна всего 2,05.

III. Строгий стиль в музыке. Построение конечного алфавита музыкального текста

Строгий стиль[7] (С.С.) – историческое и художественно-стилистическое понятие, относящееся к хоровой полифонической музыке эпохи Ренессанса (XV–XVI вв.). Оно охватывает широкий круг явлений и не имеет ясно очерченных границ и относится к творчеству композиторов разных европейских школ. Нормы строгого письма были теоретически сформулированы Дж. Царлино (1517–1590) в трактате «Гармонические установления».

Образный строй музыки С.С. – возвышенный и сдержанный, отсутствуют контрасты, резкие повороты движения, нарастания и спады.

Большая часть произведений С.С. предназначена для хора *a capella*; особенности мелодики и метроритма в музыке этого направления во многом определяются его вокально-хоровой природой. Композиторы тщательно устраняют из своих произведений все, что могло бы помешать естественному движению голоса, непрерывному развертыванию мелодической линии, все, что кажется слишком резким, способным привлечь внимание к частности, к детали. Очертания мелодий плавны, в мелодической линии отсутствуют скачки на трудно интонируемые диссонирующие и широкие интервалы, преобладает постепенное движение, а скачки уравновешиваются движением в обратном направлении. Для ритмической организации не типично соседство звуков, значительно отличающихся по длительности; с целью достижения ритмической ровности из двух слигованных нот вторая обычно или равна первой, или короче ее вдвое и т. д.

Относительное стилистическое единство музыки эпохи С.С., простота мелодико-гармонических и ритмических норм позволяют изложить основы контрапункта в виде сравнительно небольшого числа точных правил и формул.

Подобное изложение приведено в учебнике полифонии В.П. Фраёнова[8].

Как уже было сказано выше, главное препятствие в сравнении МТ и ВТ заключается в большой сложности представления МТ в виде конечной совокупности конечных последовательностей символов (букв «музыкального алфавита»). То обстоятельство, что сформулированные в учебнике полифонии В.П. Фраёнова правила С.С. адаптированы для написания учебных работ (студентами музыкальных учебных заведений), а следовательно, несколько упрощены и формализованы, способствует значительному облегчению данной задачи.

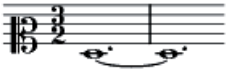
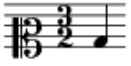
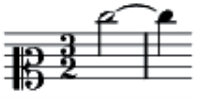
В.П. Фраёновым указаны строгие ограничения на употребляемые высоты и длительности нот, заданы правила связывания нот. Эти ограничения легли в основу предлагаемого «музыкального алфавита».

Различных вариантов звуковысотного положения ноты может быть 14, вариантов ее длительности для размера $^3/_2$ – 13 (две залигованные ноты будем здесь считать отдельной самостоятельной длительностью).

Предлагается использовать $13 \times 14 = 182$ символа (комбинации всех возможных длительностей со всеми возможными высотами) в качестве «букв» составляемого алфавита.

Примеры символов полученного алфавита приведены в табл. 2

Таблица 2

		Высота	Длительность	Комментарий
1		Фа малой октавы	Целая нота с точкой (1 ½ целой), слигованная с целой нотой с точкой (1 ½ целой)	Общая длительность: 3 целых
50		Си малой октавы	Четвертная нота	
175		Ми второй октавы	Половинная нота, слигованная с четвертной нотой	Общая длительность: ¾ целой

IV. Применение модели фон Неймана к формальному представлению музыкального текста

Для применения модели фон Неймана нулевого приближения к полученному музыкальному алфавиту (мощность $Z = 182$) достаточно воспользоваться формулой $\sum_{k=1}^{l} Z^k$ для количества слов длины, не превышающей l.

Для применения же к нему модели фон Неймана первого приближения необходимо составить матрицу запретных биграмм. Поскольку составление матрицы 182×182 представляет собой весьма трудоемкий и длительный процесс, то были найдены пути упрощения решения данной задачи. Выяснилось, что формализованные правила С.С., регламентирующие последование двух длительностей нот, с одной стороны, и последование двух звуковысотных положений нот – с другой, почти не коррелируют между собой (единственный случай зависимости длительности и высоты – запрет на скачки с участием восьмых нот).

Поэтому были составлены отдельно две матрицы запретных биграмм: одна – для соседних длительностей, другая – для соседних высотных положений нот. Потом была написана программа для составления матрицы 182×182 (на основе этих двух матриц и с учетом указанного выше запрета) и произведены вычисления количества слов в модели первого приближения.

В табл. 3 приведены результаты вычислений для количества мелодий в модели фон Неймана нулевого и первого приближений. Количество запретных пар символов в МТ С.С. достаточно велико,

благодаря чему значительно сокращается число «слов» (мелодий) в модели первого приближения.

V. Сравнительный анализ полученных моделей

На данном этапе работы получены числовые характеристики для модели фон Неймана нулевого и первого приближений, примененной к ВТ и МТ.

Для русского языка матрица запретных биграмм была составлена на базе словаря с помощью методов автоматической обработки текста, для МТ же подобная матрица была составлена вручную (с использованием лишь вспомогательной программы) на основе формализованных правил С.С. Трудоемкость составления для МТ матриц, учитывающих правила запретных последовательностей из большего числа символов (матрица запретных трехграмм, четырехграмм и т. д.), вынуждает нас ограничиться в данном исследовании рассмотрением моделей фон Неймана лишь нулевого и первого приближений.

Таблица 3

Количество слов (мелодий) в модели фон Неймана нулевого и первого приближений для музыкального текста строгого стиля

Длина	$S_0(I)$	$S_1(I)$	$\lg(S_0(I)/S_1(I))$
1	182	182	0,00
2	33124	6261	0,72
3	6061692	276082	1,34
4	1103261068	11727028	1,97
5	$2{,}00794 \cdot 10^{11}$	498582253	2,61
6	$3{,}65444 \cdot 10^{13}$	21183472665	3,24
7	$6{,}65109 \cdot 10^{15}$	$9{,}00052 \cdot 10^{11}$	3,87
8	$1{,}2105 \cdot 10^{18}$	$3{,}82406 \cdot 10^{13}$	4,50
9	$2{,}20311 \cdot 10^{20}$	$1{,}62473 \cdot 10^{15}$	5,13
10	$4{,}00965 \cdot 10^{22}$	$6{,}90299 \cdot 10^{16}$	5,76
11	$7{,}29857 \cdot 10^{24}$	$2{,}93287 \cdot 10^{18}$	6,40
12	$1{,}32816 \cdot 10^{27}$	$1{,}24609 \cdot 10^{20}$	7,03
13	$2{,}41725 \cdot 10^{29}$	$5{,}29423 \cdot 10^{21}$	7,66

Окончание таблицы 1

Длина	$S_0(I)$	$S_1(I)$	$\lg(S_0(I)/S_1(I))$
14	$4{,}39939 \cdot 10^{31}$	$2{,}24936 \cdot 10^{23}$	8,29
15	$8{,}00689\,10^{33}$.	$9{,}55682 \cdot 10^{24}$	8,92
16	$1{,}45725 \cdot 10^{36}$	$4{,}0604 \cdot 10^{26}$	9,55
17	$2{,}6522 \cdot 10^{38}$	$1{,}72514 \cdot 10^{28}$	10,19
18	$4{,}82701 \cdot 10^{40}$	$7{,}32958 \cdot 10^{29}$	10,82
19	$8{,}78515 \cdot 10^{42}$	$3{,}11411 \cdot 10^{31}$	11,45
20	$1{,}5989 \cdot 10^{45}$	$1{,}32309 \cdot 10^{33}$	12,08
21	$2{,}90999 \cdot 10^{47}$	$5{,}6214 \cdot 10^{34}$	12,71
22	$5{,}29619 \cdot 10^{49}$	$2{,}38836 \cdot 10^{36}$	13,35
23	$9{,}63906 \cdot 10^{51}$	$1{,}01474 \cdot 10^{38}$	13,98
24	$1{,}75431 \cdot 10^{54}$	$4{,}31132 \cdot 10^{39}$	14,61
25	$3{,}19284 \cdot 10^{56}$	$1{,}83174 \cdot 10^{41}$	15,24

При условии существования методов автоматической обработки и наличия баз данных МТ интересующего нас стиля, адаптированных для подобной обработки, было бы возможным провести гораздо более полный и глубокий сравнительный анализ МТ с ВТ.

Тем не менее и в рамках предложенных моделей могут быть получены результаты, представляющие определенный интерес.

Как было отмечено выше, процент запретных биграмм в модели фон Неймана первого приближения для МТ значительно выше, чем для ВТ (для ВТ – 233 из 1089, что составляет около 20%, для МТ – 26863 из 33124, что составляет около 80%). Это обусловлено, с одной стороны, уже упоминавшейся особенностью МТ, а именно его художественной спецификой, естественно требующей более жесткой фильтрации сочетаний символов; с другой стороны, это следствие строгости правил самого С.С., вызванной как его вокально-хоровой природой, так и чертами художественно-эстетических воззрений эпохи.

Указанная строгость правил приводит к значительному сокращению количества слов в модели первого приближения в сравнении с моделью нулевого приближения. Тем не менее, благодаря существенно большей мощности алфавита МТ, оно все же остается гораздо большим, чем для ВТ.

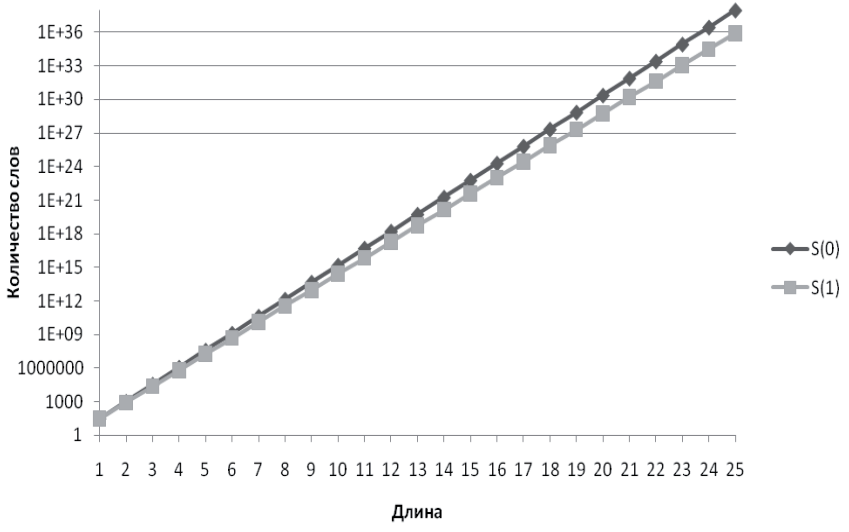


Рис. 1. Количество слов в модели фон Неймана нулевого и первого приближений для русского языка

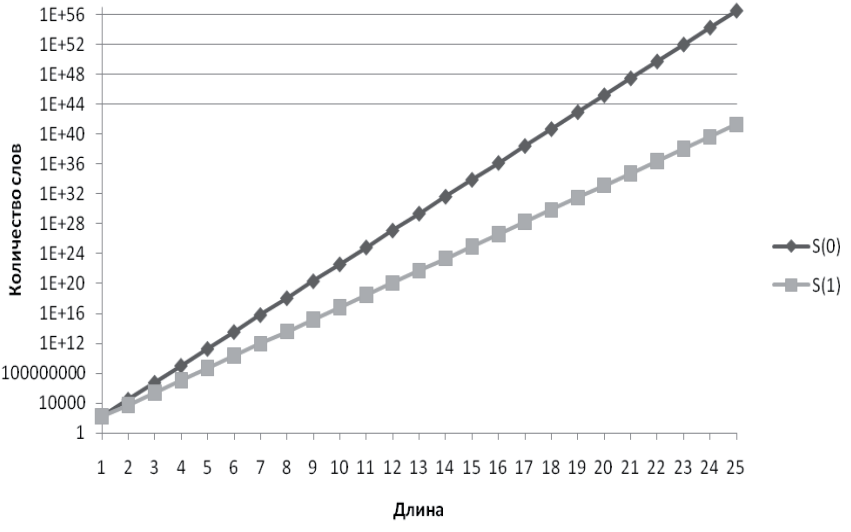


Рис. 2. Количество слов (мелодий) в модели фон Неймана нулевого и первого приближений для музыкального текста строгого стиля

Обозначенные особенности отражены на приведенных ниже графиках (рис. 1–3). Для удобства во всех случаях используется логарифмическая шкала, так как исследуется степенная функция. На первых двух из них (рис. 1–2) сравнивается количество слов в моделях разных приближений для МТ и ВТ.

На третьем графике (рис. 3) отражено сравнение количества слов в модели первого приближения для обоих видов текстов.

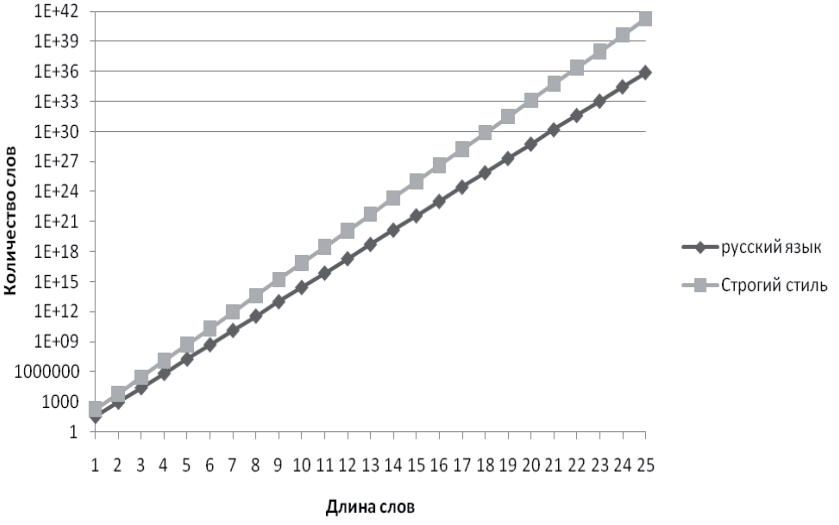


Рис. 3. Количество слов в модели фон Неймана первого приближения для русского языка и строгого стиля

Рассмотрим теперь полученные модели в контексте оценки пропускной способности канала передачи информации. Пропускная способность канала передачи информации равна $C = V_{\max} H$. Так как V зависит лишь от физических характеристик линии связи, то необходимо сравнить только H – энтропию сообщений.

В моделях нулевого приближения (отсутствуют взаимосвязи между символами) и при условии, что разная вероятность появления символов не учитывается, применима формула $H = \log_2 m = -\log_2 P$, где m – количество символов алфавита, $P = 1/m$ – вероятность появления каждого символа.

Для ВТ $H = \log_2 33 \approx 5{,}04$; для МТ $H = \log_2 182 \approx 7{,}51$.

Таким образом, в модели фон Неймана нулевого приближения пропускная способность канала музыкальной коммуникации больше, но только за счет большей мощности алфавита.

В моделях первого приближения (символы коррелированы друг с другом) необходимо использовать формулу для условной энтропии

$$H(A'/A) = -\sum_{i=1}^{m} P(a_i) \sum_{j=1}^{m} P(a'_j/a_i) \log P(a'_j/a_i),$$

где $P(a'_j/a_i)$ – условная вероятность появления символа a_j после символа a_i.

Полученные данные таковы: для русского языка $H_r \approx 4{,}6398$, для МТ С.С. $H_m \approx 4{,}6018$.

Соответственно, и коэффициент избыточности $r = (H_1 - H)/H_1 = 1 - H/\log m$ для русского языка $r_r \approx 0{,}08$, для МТ С.С. $r_m \approx 0{,}38$.

Какие можно сделать выводы из полученных результатов? С одной стороны, проведенные вычисления констатируют факт большей пропускной способности канала вербальной коммуникации. Тем не менее разница H_r и H_m не столь велика, отличие появляется лишь в сотых долях. С другой стороны, результаты вычислений, проведенные на модели МТ С.С., представляют собой *оценку снизу* всех аналогичных результатов для подавляющего большинства музыкальных текстов. Это происходит вследствие того, что МТ С.С. обладает, во-первых, минимальным алфавитом, а во-вторых, подчиняется максимально строгим правилам, регламентирующим запретные последовательности нот.

Таким образом, даже для одноголосия вполне оправдана гипотеза о большей пропускной способности канала музыкальной коммуникации.

VI. Примерная оценка повышения пропускной способности для многоголосного канала музыкальной коммуникации

При попытке обратиться к рассмотрению нескольких звучащих голосов сразу возникает вопрос: в рамках каких моделей рассматривать многоголосный МТ?

Если не учитывать правил вертикального соединения в выбранном музыкальном стиле, можно было бы просто умножить параметры, полученные для одноголосия, на соответствующее число голосов. Однако упомянутые правила для строгого стиля достаточно детерминированы и обеспечивают отсечение большого числа вариантов из всех возможных вертикальных соединений. Поэтому на примере двухголосия рассмотрим модель, учитывающую эти правила.

Предлагается ввести новый алфавит, каждым символом которого будет некоторое вертикальное соединение, содержащее две ноты. Причем относительное временное расположение этих двух нот относительно друг друга может быть любым, лишь бы существовал момент их одновременного звучания. Таким образом, будем иметь сочетания, подобные следующим: «вторая четверть целой плюс четверть», «третья половина целой с точкой плюс половина», а в двухголосном нотном тексте новый символ будет наступать каждый раз при изменении одного из голосов.

Составим примерную таблицу (табл. 4) для количества таких сочетаний. Для упрощения задачи не будем учитывать существования восьмых долей, а длительности остальных нот выразим в числе содержащихся в них четвертей (при этом ранее дифференцировавшиеся длительности, содержащие одинаковое число четвертей, теперь будем считать одинаковыми). Содержимое каждой ячейки таблицы 4 отражает количество вариантов сочетаний длительностей соответствующей длины.

Таким образом, имеем 328 сочетаний. При этом здесь учтены сочетания только длительностей нот. По звуковысотному положению каждое сочетание может быть повторено при фиксированной высоте одной из нот по крайней мере 4 раза (по числу консонирующих созвучий, на употребление которых в С.С. не наложены ограничения; правила же употребления диссонансов достаточно сложны, и мы их не будем учитывать). Если же фиксировать указанную ноту на разных высотах (при этом интервал должен помещаться в заданный диапазон), то получим еще не менее пяти возможностей, т. е., в общей сложности 20 различных сочетаний.

В результате имеем следующую нижнюю оценку мощности алфавита для двухголосия: $328 \times 20 = 6560$ символов.

Таблица 4

	1	2	3	4	6	8	10	12
1	1	2	3	4	6	8	10	12
2	2	1	2	3	5	7	9	11
3	3	2	1	2	4	6	8	10
4	4	3	2	1	3	5	7	9
6	6	5	4	3	1	3	5	7
8	8	7	6	5	3	1	3	5
10	10	9	8	7	5	3	1	3
12	12	11	10	9	7	5	3	1

Таким образом, мы видим, что уже для двухголосия, учитывающего все правила С.С., мощность алфавита увеличивается на порядок. Благодаря этому увеличивается и энтропия сообщений, состоящих из символов этого алфавита. И хотя с возрастанием числа голосов возрастает и строгость правил, регламентирующих последовательные сочетания нот, но все же не настолько, чтобы оказать значительное влияние на количество слов в моделях следующих приближений.

Для модели нулевого приближения имеем: $H = \log_2 6560 \approx 16{,}01$ (тогда как для ВТ $H = \log_2 33 \approx 5{,}04$).

VII. Семантика музыкального текста

Ввиду малой изученности МТ как самостоятельного математического феномена логично рассматривать его семантические аспекты также в соотнесении с ВТ.

Структурная сложность МТ отражается и на семантическом уровне. ВТ допускает разложение на отдельные структурные семантически неделимые единицы – слова, каждое из которых обладает своим собственным ограниченным семантическим полем. В МТ же подобное деление приводит нас к отдельным звукам, каждый из которых не обладает собственной семантикой, а с необходимостью включен в окружающую музыкальную ткань.

Очевидно, что и в ВТ семантика, например, предложения не будет равна простой сумме семантических значений образующих его слов, но речь идет о принципиальной невозможности выделить в МТ некие элементарные структурно детерминированные единицы, обладающие собственной семантикой. В отдельных случаях для музыкальных фраз, фигур и т. п. возможно ограничение области допустимой семантической интерпретации (например, барочные риторические фигуры – определенные музыкальные обороты, призванные подчеркнуть аффект произведения или данного его фрагмента), но и здесь имеет место не точная мелодико-ритмическая формула, а лишь обобщенный структурный тип (например, «фигура креста»), который может быть выражен абсолютно разными музыкальными средствами.

При этом необходимо различать объективную семантику, присущую самой информации, и субъективную (прагматическую) семантику, зависящую от воспринимающего субъекта. По определению А.Е. Барановича[9], «*объективная семантика* информации характеризует информационные формы существования материальных систем объективной реакции и взаимосвязана с формой, струк-

турой и организацией материальных систем. <...> В свою очередь, *семантика субъективная* (прагматическая) интерпретируется как динамический информационный образ объективной семантики (информации материальной системы "внешнего мира"), инициализированный в подсистеме знаний воспринимающей интеллектуальной системы».

Тем не менее оба эти аспекта семантики неотделимы друг от друга: «...объективная информация неотрывна от субъективной интерпретации исследователя, то есть от прагматического отношения информации к субъекту. Тем более, если речь идет об этапах восприятия и распознавания информации интеллектуальной системой, формирования ее "смысла" (однокоренное с "мыслью") и "значения" и, далее, фиксации субъективной интерпретации в вербальной форме в подсистеме знаний антропных интеллектуальных систем. Таким образом, в предметной области существования и семантической коммуникации интеллектуальной системы объективная семантика информации неотъемлема от ее прагматической составляющей».

Для дальнейшего изложения воспользуемся предложенным А.Е. Барановичем определением прагматической семантики[10]: «Семантика информации есть индивидуальная характеристика результата информационного взаимодействия интеллектуальной системы с материальной системой – внешней средой, связывающая формируемый в подсистеме знаний интеллектуальной системы как модели объективной реальности информационный образ взаимодействующей материальной системы (семантический образ) с ее воспринимаемым информационным образом».

Таким образом, мы имеем два взаимодействующих объекта: само произведение, текст, как модель содержания, вложенного в него автором, и воспринимающий субъект, соотносящий данное произведение с существующей в его сознании моделью реальности.

Очевидно, что для МТ множество различных вариантов восприятия будет гораздо больше, чем для ВТ. Вообще, семантическую неоднозначность можно считать одним из критериев художественности текста. Кроме того, для музыкального произведения существенно наличие в цепочке «автор текста – акцептор текста» дополнительного звена – «интерпретатор» (исполнитель), что создает еще одну точку ветвления семантики.

Тем не менее континуальность множества возможных интерпретаций и вариантов восприятия музыкального произведения вовсе не означает абсолютной допустимости его произвольного толкования. В работе советского музыковеда Г. Головинского «О вариантности восприятия музыкального образа»[11] автор на основе эксперимен-

тальных данных показывает, что существуют некие архетипы сознания, обуславливающие общность типа слушательской реакции на данное произведение.

Из факта же объективного существования подобных различных типов реакции следует потенциальная возможность описания продуцирующих их моделей семантики музыкального произведения в терминах точных наук.

Выводы

Анализ полученных результатов приводит нас к следующим положениям.

• При сравнении ВТ и МТ можно видеть, что алфавиты последнего обладают гораздо большей мощностью, что при отсутствии других параметров позволяет говорить о большей энтропии сообщений, использующих музыкальный алфавит, и следовательно, о большей пропускной способности музыкального канала связи по сравнению с вербальным: $C_m > Cv$.

• Однако МТ в силу своей художественной природы обладает и гораздо более строгим набором запретов на сочетания символов, что приводит к детерминированности следующего символа исходя из предыдущих, а следовательно, к уменьшению энтропии и понижению пропускной способности канала музыкальной коммуникации.

• От сочетания этих двух параметров – мощности алфавита и строгости правил – и зависит в конечном итоге пропускная способность музыкального канала коммуникации. И если для некоторых моделей музыкальных стилей (как, например, для рассмотренного одноголосного С.С., представляющего собой своеобразную оценку снизу по энтропии для всех музыкальных стилей) музыкальный контекст проигрывает вербальному, то для других (например, для додекафонной техники, энтропию которой можно считать, наоборот, оценкой сверху для любого стиля) преимущество музыкального контекста очевидно.

• Для решения задач, обозначенных в разделе VII настоящей работы, необходимо последовательное исследование модели семантики в применении к МТ – на изучение данной области направлены в настоящий момент основные усилия автора.

Автор приносит глубокую благодарность проф. А.Е. Барановичу за постановку задачи, ценные рекомендации и помощь при проведении исследований.

Список аббревиатур

ВТ – вербальный текст
МТ – музыкальный текст
С.С. – строгий стиль

Примечания

1 См.: ФГУ «НИИ РИНКЦЭ». База данных «VAK». Паспорт специальности 05.13.17. [Электронный ресурс]. [М., 2010]. URL: http://www.extech.ru/library/spravo/vak/vak_pasport/ne_abonent/add.php?kod=05.13.17&str=1&page=4&kod1=5 (дата обращения: 20.12.2010).

2 См.: *Адамов Г.Б.* Тайна двух океанов. М.: ЭКСМО, 2007; *Баранович А.Е.* Постнеклассические аспекты информационной безопасности интеллектуальных систем // 25 лет ИИНТБ. М.: РГГУ, 2010 (в печати).

3 См.: *Гуров И.П.* Основы теории информации и передачи сигналов: Электронный учебник по дисциплине «Теория информации и передачи сигналов» [Электронный ресурс]. [СПб.: СПбГУ ИТМО, кафедра компьютерных технологий]. URL: http://de.ifmo.ru/bk_netra/page.php?tutindex=11 (дата обращения: 20.12.2010).

4 См.: *Баранович А.Е.* Семиотико-хроматические гипертопографы. Введение в аксиоматическую теорию: информационный аспект. М.: ГШ ВС РФ, 2003.

5 Там же.

6 См.: *Зализняк А.А.* Грамматический словарь русского языка [Электронный ресурс]. [М.: АСТ-Пресс Книга, 2010]. 720 с.

7 См.: Музыкальная энциклопедия: В 6 т. / Гл. ред. Ю.В. Келдыш. Т. 5: Симон – Хейлер М.: Советская энциклопедия, 1981. 1056 с.

8 См.: *Фраёнов В.П.* Учебник полифонии. М.: Музыка, 1987. 207 с.

9 См.: *Баранович А.Е.* Семантические аспекты информационной безопасности: концентрация знаний // Вестник РГГУ. Сер. «Информатика. Защита информации. Математика» / Рос. гос. гуманит. ун-т (в наст. сб.).

10 См.: *Баранович А.Е.* Структурное метамоделирование телеологических информационных процессов в интеллектуальных системах. М.: ГШ ВС РФ, 2002; *Он же.* Дидактические материалы к специальному курсу «Введение в информациологию и ее специальные приложения». М.: РГГУ, 2010.

11 См.: *Головинский Г.* О вариантности восприятия музыкального образа // Восприятие музыки. М.: Музыка, 1980. С. 127–140.

А.С. Малкова

ФОРМАЛЬНОЕ РАЗРЕШЕНИЕ ПРОБЛЕМЫ ПРОТИВОРЕЧИВОСТИ ОЦЕНОК В ЦЕННОСТНЫХ ВЫСКАЗЫВАНИЯХ (НА МАТЕРИАЛЕ РУССКИХ ПОСЛОВИЦ)

В статье дается описание разработанной семантической модели пословичного высказывания. Пословица интерпретируется как особая семантическая единица, содержащая специфические (ценностно-ориентированные) знания о мире. Описываются регулярные случаи противоречивой оценки одних и тех же явлений (позитивной и негативной) и способы их формального разрешения в модели.

Ключевые слова: представление знаний, ценностные высказывания, пословица.

Введение

В работах[1,2,3] был предложен подход к формальному представлению семантики ценностных высказываний (на материале русских пословиц), представляющий собой формализацию методов структурного анализа, развивавшихся в рамках специальных дисциплин гуманитарного направления: классической филологии[4], структурной антропологии[5, 6], когнитивной лингвистики[7]. На основе данного подхода была разработана формальная модель семантики пословичного высказывания. В настоящей статье кратко излагается структура модели и обсуждаются полученные в результате анализа текстов из экспериментальной выборки способы разрешения в модели одной из наиболее *ярких* проблем: проблемы внутренней противоречивости пословичного фонда.

Действительно, в пословицах часто встречаются противоречащие друг другу утверждения либо амбивалентное отношение к одному и тому же явлению, например:

1. *По труду и честь*. vs. *Работа дураков любит*.

2. *Ученье – свет, неученье – тьма*. vs. *Многие знания – многие печали*.

3. *Своя ноша не тянет*. vs. *Чужая ноша не тянет*.

4. *Кто рано встает, тому Бог дает*. vs. *Трудом праведным не наживешь палат каменных*.

5. *Учиться никогда не поздно*. vs. *Горбатого могила исправит*.

6. *Сам погибай, а товарища выручай*. vs. *Своя рубаха ближе к телу*.

7. *Живи тихо – не будет тебе лиха*.vs. *Кто живет тихо, тому от людей лихо*.

Этот список можно продолжить. Внутренняя противоречивость представляет собой слабо исследованную, но чрезвычайно интересную особенность пословичного фонда. При построении модели мы неизбежно сталкиваемся с необходимостью дать этому явлению формальную интерпретацию.

В фольклористке существует два взгляда на проблему противоречивости. Согласно одному из них[8] пословицы являются *знаками ситуаций* (в контексте которых они употребляются), поэтому наличие противоречий в пословичном фонде не только допустимо, но и закономерно: пословицы отражают все многообразие жизненных перипетий.

Согласно другому, более традиционному взгляду[9,10,11], пословицы не просто иллюстрируют бытовые ситуации, но и отражают *ценностные установки* народа – носителя культуры. Несмотря на наличие в фонде противоречащих друг другу высказываний, одна оценка интуитивно ощущается этически правильной, «моральной», она представлена большим количеством высказываний, в то время как противоположная ей кажется «аморальной».

В модели вторая точка зрения (традиционная) получила структурное обоснование. В результате анализа текстов из экспериментальной выборки были выявлены и формально описаны правила обработки регулярных противоречий. Они рассматриваются в настоящей статье.

1. Формальная модель семантики пословицы

Кратко изложим структуру модели. Одновременно с формальной записью был разработан графический язык представления, призванный облегчить визуальное восприятие структурных схем пословиц. Между графическими схемами и формальными записями существует взаимно-однозначное соответствие.

1.1. Базовые объекты и отношения модели

Базовыми ***объектами*** модели являются единицы, имеющие бинарную структуру, или *оппозиции*. Это понятия, характеризующие человека или некоторые внешние обстоятельства с двух сторон – позитивно и негативно, например: добродетель–порок, успех–неудача. Оппозиция представляет собой упорядоченную пару

$$o = \langle p, n \rangle \in O,$$

где p – понятие с позитивной оценкой;

n – понятие с негативной оценкой.

Оппозиции можно разбить на два класса с помощью сюръективного отображения: class: $O \rightarrow M$, где $M = \{\text{man, univ}\}$.

$M_{man} = \{o \mid \text{class}(o) = \text{man}\}$ – оппозиции, характеризующие поступки, черты характера человека. $M_{man} = \{\langle$добродетель, порок$\rangle, \langle$знания, глупость$\rangle, \langle$труд, безделье$\rangle, \langle$помощь, вред$\rangle, \langle$воля, покорность$\rangle\}$.

$M_{univ} = \{o \mid \text{class}(o) = \text{univ}\}$ – оппозиции, характеризующие внешние обстоятельства, на которые человек не может непосредственно повлиять. $M_{univ} = \{\langle$успех, неудача$\rangle, \langle$доход, убыток$\rangle, \langle$удовольствие, страдание$\rangle, \langle$покой, работа$\rangle, \langle$статус, презрение$\rangle, \langle$дружба, одиночество$\rangle\}$.

В графической записи оппозиции описываются двумя расположенными одна над другой окружностями. Верхняя окружность соответствует позитивному элементу оппозиции, а нижняя – негативному.

Рис 1. Графическое представление оппозиции

Базовым ***отношением*** модели, или *звеном*, является маркированное отношение между двумя оппозициями. Оно описывается кортежем

$$\langle\langle t1, t2\rangle, o1, o2, r, c\rangle,$$

где $t1, t2 \in T$ – акторы (действующие лица) ценностного высказывания, $T = \{$человек, мир$\}$;

$o1, o2 \in O$ – оппозиции;

$r=\langle\omega,\alpha\rangle \in R$ – семантический маркер отношения, или уподобление (подробнее об уподоблениях см. ниже);

$c \in \sum\times\sum\times\sum$, где $\sum=\{-1, +1\}$ – конфигурация связи, т. е. стандартное сочетание ±-оценок оппозиций, входящих в отношение, с результирующей оценкой всего звена (подробнее о конфигурациях также см. ниже).

Таким образом, *звено* – это отношение между двумя оппозициями, основанное на некотором уподоблении и обладающее результирующей этической оценкой. В графической записи две оппозиции изображаются рядом и каждая из них соотносится с одной из частей уподобления $r=\langle\omega, \alpha\rangle$.

Рис. 2. Графическое представление отношения

Уподобление $r=\langle\omega,\alpha\rangle \in R$ описывает основание, по которому противопоставляются две оппозиции в рамках отношения. $R=$ {⟨временно, в итоге⟩, ⟨форма, содержание⟩, ⟨мнимо, реально⟩, ⟨материально, духовно⟩, ⟨желаемое, доступное⟩, ⟨часть, целое⟩, ⟨иногда, всегда⟩, ⟨некто, сам⟩, ⟨сфера, акция⟩}. Семантика уподоблений восходит к когнитивному механизму мифологического отождествления, детально рассмотренному в работе[12].

Типы акторов $t1, t2 \in T$ определяют, к какому из классов M_{man}, M_{univ} принадлежат оппозиции *o1*, *o2* соответственно. Поскольку существует два типа акторов $T=$ {человек, мир}, возможно четыре варианта их сочетания в звене.

1. $o1 \in M_{man}$, $o2 \in \mathrm{M}_{\mathrm{univ}}$: звено описывает влияние поступков, черт характера человека на его судьбу, отношение к нему других людей, например: *кто трудится, будет жить в достатке*.

2. $o1 \in M_{univ}$, $o2 \in M_{man}$: звено описывает влияние внешних обстоятельств на поступки человека, необходимость совершать вынужденные действия, например: *что пропало, того уже не вернуть*.

3. $o1, o2 \in M_{man}$: звено устанавливает приоритеты между чертами характера человека или внутри коллектива, например: *главное не внешность, а сущность человека*.

4. $o1, o2 \in M_{univ}$: звено устанавливает приоритеты между внешними обстоятельствами, например: *дружба важнее богатства*.

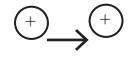
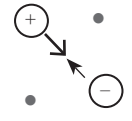
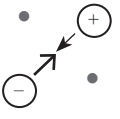
Таким образом, в высказываниях, где оппозиции принадлежат к разным классам, полученное отношение может быть охарактеризовано как «влияние», где к одному – как «предпочтение».

Последний элемент структурного описания звена – конфигурация $c = \langle \sigma 1, \sigma 2, \sigma \rangle \in \sum \times \sum \times \sum$ – описывает стандартное сочетание ±-оценок оппозиций, входящих в звено, с результирующей этической оценкой всего звена $\sum = \{-1, +1\}$. Положительная результирую-щая оценка звена соответствует генеральной идее одобрения или оправдания, отрицательная – идее несправедливости или осуждения; ±-оценки оппозиций указывают на то, какая из частей оппозиции – положительная или отрицательная – задействуется в звене.

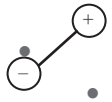
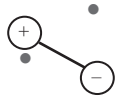
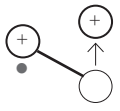
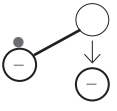
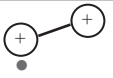
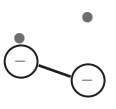
Конфигурация является независимым семантическим элементом отношения и несет самостоятельную смысловую нагрузку. В модели всего выявлены десять различных видов конфигураций: четыре для отношений типа «влияние» и шесть для отношений типа «предпочтение». Они могут быть объединены в пары (у двух конфигураций в паре все три оценки $\sigma 1$, $\sigma 2$ и σ противоположны). Графические схемы конфигураций и примеры интерпретаций приведены в табл. 1 (в примерах для иллюстрации используются оппозиции ⟨труд, безделье⟩, ⟨доход, убыток⟩ и уподобление ⟨часть, целое⟩). В графической схеме конфигураций для отношений типа «предпочтение» расстояние между полюсами у доминирующей оппозиции больше, чем у субдоминантной.

Таблица 1

Графические схемы конфигураций

Оценки			Граф. схема		Пример интерпретации
$\sigma 1$	$\sigma 2$	σ	а)	б)	
ТИП «ВЛИЯНИЕ»					
+1	+1	+1			а) кто *трудится*, будет иметь *доход*
–1	–1	+1			б) кто *бездельничает*, потерпит *убыток*
+1	–1	–1			а) некто *трудится*, а терпит *убытки*
–1	+1	–1			б) некто *бездельничал*, а имеет *доход*

Окончание таблицы 1

Оценки			Граф. схема		Пример интерпретации
σ1	*σ2*	*σ*	а)	б)	
ТИП «ПРЕДПОЧТЕНИЕ»					
–1 +1	+1 –1	+1 –1			а) пусть *часть* плохая, зато *целое* хорошее б) хотя *часть* хорошая, да *целое* плохое
+1 –1	–1 +1	+1 –1			а) когда *целое* плохое, хорошая *часть* – уже хорошо б) плохая *часть* портит хорошее *целое*
+1 –1	+1 –1	+1 –1			а) *целое* хорошее и *часть* хороша б) *целое* плохое и *часть* плоха

Рассмотрим два примера звеньев: их формальную запись, графическую схему и семантическую интерпретацию. В пояснениях к формулам курсивом будем обозначать уподобления, обычным текстом – оппозиции, полужирным – конфигурации; в квадратных скобках будем размещать дополнительные поясняющие слова.

Пример 1. *Человек с виду добродетельный, в действительности может оказаться дурным.*

⟨⟨человек, человек⟩, ⟨добродетель, порок⟩, ⟨добродетель, порок⟩, ⟨форма, содержание⟩, ⟨+1,–1,–1⟩⟩, т. е. ‘**хотя** *форма* [человека] добродетельная, **да** *содержание* порочное’.

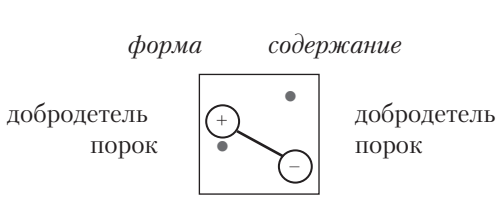


Рис. 3. Графическая схема звена из примера 1

Пример 2. *Кто трудится, будет жить в достатке.*

⟨⟨человек, мир⟩, ⟨труд, безделье⟩, ⟨доход, убыток⟩, ⟨человек, мир⟩, ⟨+1,+1,+1⟩⟩, т. е. *'если* труд, *то* доход'.

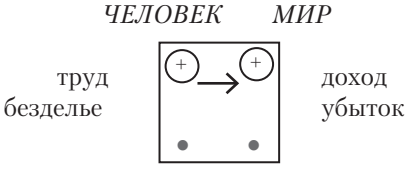


Рис. 4. Графическая схема звена из примера 2

1.2. Модель реальности. Полная схема пословицы

Как было показано выше, пословицы отражают взаимодействие человека с окружающим миром. В них отражены четыре основных вида утверждений:

1. ЧЕЛОВЕК → МИР: влияние поступков, черт характера человека на его судьбу, отношение к нему других людей.
2. МИР → ЧЕЛОВЕК: влияние внешних обстоятельств на поступки человека; необходимость совершать вынужденные действия.
3. ЧЕЛОВЕК > ЧЕЛОВЕК: установление приоритетов, зависимостей между чертами характера человека или внутри коллектива.
4. МИР > МИР: установление приоритетов, зависимостей между внешними обстоятельствами.

Данные четыре вида утверждений могут быть объединены в ***полной схеме*** пословицы, включающей *три* звена и представляющей собой упорядоченную тройку

Prov = ⟨⟨⟨*t11*, *t12*⟩, *o11*, *o12*, *r1*, *c1*⟩,
⟨⟨*t1*, *t2*⟩, *o1*, *o2*, *r*, *c*⟩,
⟨⟨*t21*, *t22*⟩, *o21*, *o22*, *r2*, *c2*⟩, где

t1 = *t11* = *t12* = человек, *t2* = *t21* = *t22* = мир, *o1* = *o12*, *o2* = *o22*.

Графическая схема полной модели пословицы включает схемы трех звеньев:

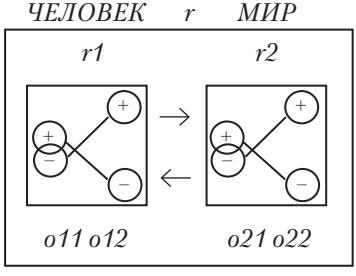


Рис. 5. Графическая схема полной модели пословицы

Первое и третье звенья построены по типу «предпочтение» и описывают приоритеты внутри каждой из сфер МИР и ЧЕЛОВЕК. Второе (центральное) звено основано на отношении типа «влияние». Оно устанавливает причинно-следственную связь между двумя сферами, направленную либо в одну, либо в другую сторону. Таким образом, все акторы первого звена и первый актор второго звена – **человек**, второй актор второго звена и все акторы третьего звена – **мир**. В полной модели пословицы некоторые звенья могут оставаться пустыми (незадействованными).

Пример 3. *Отольются волку овечкины слезки* (т. е. дурной человек временно может оставаться безнаказанным, но рано или поздно его настигнет возмездие.

Семантика пословицы складывается из двух звеньев.

Схема 1: *порок наказывается*

⟨⟨человек, мир⟩, ⟨добродетель, порок⟩, ⟨удовольствие, страдание⟩, ⟨человек, мир⟩, ⟨–1, –1, –1⟩⟩, т. е. '**если** порок, **то** страдание'

Схема 2: *безнаказанное положение временно*

⟨⟨мир, мир⟩, ⟨удовольствие, страдание⟩, ⟨удовольствие, страдание⟩, ⟨временно, в итоге⟩, ⟨+1,–1,–1⟩⟩, т. е. '**хотя** *временно* удовольствие, **но** *в итоге* страдания'

В данной пословице оба семантических компонента одинаково важны: и временной, и идея возмездия.

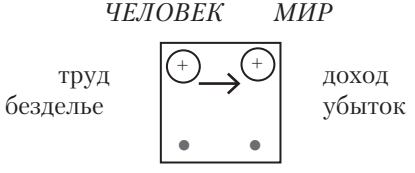


Рис. 6. Графическая схема звена из примера 2

В качестве экспериментальной выборки при верификации модели был использован словарь современных русских пословиц под ред. В.М. Мокиенко[13], содержащий 505 единиц. Выбор обусловлен тем, что при относительно сжатом объеме словарь охватывает большое тематическое многообразие пословиц, в нем представлен весь фонд, а не узкий тематический срез.

Отметим, что при формировании списков элементов модели (оппозиций, уподоблений, конфигураций) применялся «экономный» подход, при котором количество элементов минимизировалось, но каждый из них охватывал широкое семантическое поле, объединяющее множество близких понятий. Это позволило исключить возможность двоякого представления семантики, поскольку каж-

дому структурному элементу соответствует уникальный компонент смысла, не выразимый через комбинацию любых других элементов. В результате словарь был ограничен минимальным набором из 11 оппозиций (5 из класса ЧЕЛОВЕК, 6 из класса МИР), 9 типов доминирующих отношений и 5 видов конфигураций. Это позволило описать около 70% высказываний из экспериментальной выборки «хорошо» и «удовлетворительно» и еще 20% – «частично». 10% высказываний не могли быть адекватно представлены в модели.

2. Разрешение в модели проблемы противоречивости

Перейдем к рассмотрению того, как в разработанной модели могут быть представлены высказывания, содержащие противоречивую оценку одних и тех же явлений (в одних случаях – позитивную, в других – негативную).

Следует сразу исключить из рассмотрения те случаи, когда одним и тем же понятиям на поверхностном уровне соответствуют различные глубинные понятия. Например, в пословице *Много будешь знать – скоро состаришься*, судя по контексту употребления, мы имеем дело не с осуждением *знания* как такового, а с негативной оценкой *любопытства*, проявления излишнего интереса к чужим делам. Такие случаи не являются регулярными. Каждое высказывание описывается уникальной схемой, в которых, однако, могут наблюдаться частичные совпадения.

Всего в результате анализа текстов из экспериментальной выборки были выявлены два типа регулярных противоречий: 1) на базе противопоставления ЧЕЛОВЕК vs. МИР и 2) на базе противопоставления СВОЙ vs. ЧУЖОЙ.

2.1. Противоречия на базе противопоставления ЧЕЛОВЕК vs. МИР

При противоречии, основанном на противопоставлении ЧЕЛОВЕК vs. МИР, ряд явлений оценивается позитивно, если они являются добровольными действиями или результатом добровольных действий либо личных качеств человека, и негативно, если они навязаны внешними обстоятельствами, были даны незаслуженно. К таким явлениям относятся в первую очередь труд и власть.

Так, труд, с одной стороны, – это деятельность человека, направленная на повышение личного благосостояния, с другой – тяжелая вынужденная повинность. Власть оценивается позитивно,

если она является естественным признанием личных достоинств человека, и негативно, если дана человеку от рождения или получена по праву силы.

Таблица 2

Примеры противоречий на базе противопоставления ЧЕЛОВЕК vs. МИР

ЧЕЛОВЕК	МИР
Кто рано встает, тому Бог дает.	*От работы кони дохнут.* *Работа не волк.*
Победителей не судят. (≈*Все хорошо, что хорошо кончается.*)	*Победителей не судят.* (≈*Кто платит, тот заказывает музыку.*)

В модели в таких случаях, как правило, меняется преимущественная схема пословиц: ЧЕЛОВЕК → МИР характерна для позитивной оценки, МИР → ЧЕЛОВЕК – для негативной. Кроме того, в формальной схеме часто используются различные оппозиции: ⟨труд, безделье⟩, ⟨добродетель, порок⟩, относящиеся к классу ЧЕЛОВЕК, и ⟨покой, работа⟩, ⟨власть, подчинение⟩, относящиеся к классу МИР.

2.2. Противоречия на базе противопоставления СВОЙ vs. ЧУЖОЙ

Еще более сильным регулярным механизмом возникновения противоречий является противопоставление по принципу СВОЙ vs. ЧУЖОЙ. Термины СВОЙ и ЧУЖОЙ в данном случае охватывают широкий круг понятий. *Свой человек* – это близкие, друзья, родная душа, я сам, кто-то посторонний, внушающий уважение. *Свой мир* – это сфера доверительности, сочувствия, распространения интересов, влияния, уважения.

Чужое – это то, от чего говорящий мысленно отгораживается, не хочет принимать, осуждает. Враждебный человек, «неподобающий» член семьи, незнакомое, чуждое общество. Даже к самому себе, к каким-то сторонам своей личности человек, осуждая, может относиться, как к *чужому*.

Наиболее ярко это противопоставление проявляется в пословицах, которые могут трактоваться амбивалентно в зависимости от того, на чьей стороне находится фокус эмпатии, например: *Своя рубашка ближе к телу*. *Наш пострел везде поспел*. Пословицы могут

звучать либо как оправдание, похвала, если относятся к СВОИМ, либо как осуждение, ирония, если речь идет о ЧУЖИХ.

Для отражения таких семантических различий модель была расширена, и каждой из двух частей семантической структуры пословицы (МИР и ЧЕЛОВЕК) были присвоены семантические маркеры СВОЙ / ЧУЖОЙ. В результате получились четыре комбинации, приведенные в табл. 3.

Таблица 3

Маркеры СВОЙ / ЧУЖОЙ
двух частей пословичного высказывания

ЧЕЛОВЕК	МИР	Семантика
СВОЙ	СВОЙ	«*Свой человек в своем мире*» одобрение; манифестация правильного поведения; оправдание случайных промахов и слабостей
ЧУЖОЙ	СВОЙ	«*Чужой человек в своем мире*» обсуждение; справедливое наказание за дурные дела
СВОЙ	ЧУЖОЙ	«*Свой человек в чужом мире*» тяготы жизни хорошего человека во враждебном, несправедливом мире
ЧУЖОЙ	ЧУЖОЙ	«*Чужой человек в чужом мире*» вольготная жизнь дурных людей в мире, где они «правят бал»

Противоречие на поверхностном уровне возникает, когда в двух высказываниях по-разному расставлены точки зрения СВОЙ vs. ЧУЖОЙ.

Таблица 4

Примеры противоречий на базе противопоставления
СВОЙ vs. ЧУЖОЙ

«СВОЙ» ЧЕЛОВЕК	«ЧУЖОЙ» ЧЕЛОВЕК
Век живи, век учись.	*Горбатого могила исправит.*
Давши слово, держись.	*Обещанного три года ждут* (т. е. не дождешься)
Сам погибай, товарища выручай.	*Рука руку моет* (осуждение взаимопомощи)

Окончание таблицы 4

«СВОЙ» МИР	«ЧУЖОЙ» МИР
Учение – свет, неученье –тьма.	*Многие знания – многие печали.*
По заслугам честь.	*Нет пророка в своем отечестве.*
Закон есть закон.	*Закон – что дышло.*
Живи тихо – не будет тебе лиха.	*Кто живет тихо, тому от людей лихо.*

Проиллюстрируем обработку противоречий на базе противопоставления СВОЙ vs. ЧУЖОЙ на примерах формальных описаний.

Пример 1

а) Прощение *своих*, надежда на позитивные изменения.

Быль молодцу не укор. Что было, то прошло. Век живи, век учись.
⟨⟨⟨человек, СВОЙ⟩, ⟨человек, СВОЙ⟩⟩, ⟨добродетель, порок⟩, ⟨добродетель, порок⟩, ⟨временно, в итоге⟩, ⟨–1, +1, +1⟩⟩, т. е. ‘**пусть** *временно* порок, **зато** *в итоге* добродетель’.

б) Осуждение *чужих*, отрицание возможности измениться.

Горбатого могила исправит. Кому Бог ума не дал, тому кузнец не прикует.
⟨⟨⟨человек, ЧУЖОЙ⟩, ⟨человек, ЧУЖОЙ⟩⟩, ⟨добродетель, порок⟩, ⟨добродетель, порок⟩, ⟨временно, в итоге⟩, ⟨+1, –1, –1⟩⟩, т. е. ‘**хотя** *временно* добродетель [вариант: знания], **но** *в итоге* порок [вариант: невежество]’.

Пример 2

а) Положительная оценка *знания* в *своем* мире.

Учение – свет, неученье – тьма. Стреляного воробья на мякине не проведешь.
⟨⟨⟨человек, СВОЙ⟩, ⟨мир, СВОЙ⟩⟩, ⟨знания, глупость⟩, ⟨неудача, успех⟩, ⟨человек, мир⟩, ⟨+1, +1, +1⟩⟩, т. е. ‘**если** знания, **то** успех’.

б) Знания – источник страданий в *чужом* мире.

Многие знания – многие печали.
⟨⟨⟨человек, СВОЙ⟩, ⟨мир, ЧУЖОЙ⟩⟩, ⟨знания, глупость⟩, ⟨удовольствие, страдание⟩, ⟨человек, мир⟩, ⟨+1, –1, –1⟩⟩, т. е. ‘[**несправедливо, но**] **некто** знает, **и за это** страдает’.

Пример 3

а) Справедливое ограничение возможностей недостойного *чужака*.

Знай сверчок свой шесток. Не в свои сани не садись. По Сеньке шапка.
⟨⟨⟨мир, СВОЙ⟩, ⟨человек, ЧУЖОЙ⟩⟩, ⟨статус, презрение⟩, ⟨воля, покорность⟩, ⟨мир, человек⟩, ⟨–1, –1, +1⟩⟩, т. е. ‘**кто** [ЧУЖОЙ] презренный, **тот** [должен] подчиниться’.

б) Вынужденное ограничение *своих* возможностей из-за низкого положения.

Не подмажешь, не поедешь. Сила солому ломит.
⟨⟨⟨мир, ЧУЖОЙ⟩, ⟨человек, СВОЙ⟩⟩, ⟨статус, презрение⟩, ⟨воля, покорность⟩, ⟨мир, человек⟩, ⟨–1, –1, +1⟩⟩, т. е. '**кто** [СВОЙ] презренный, **тот** [вынужден] подчиниться'.

Заключение

Таким образом, в процессе разработки формальных средств описания семантики пословицы (основы изложены в более ранних работах[14, 15, 16]) были выявлены регулярные механизмы вынесения в пословицах противоречивой оценки в отношении одних и тех же явлений. Им была дана формальная интерпретация, т. е. было показано, каким образом эти механизмы могут найти отражение в структуре формальных описаний.

Примечания

1 См.: *Малкова А.С.* Разработка представления семантики ценностно ориентированных текстов в базе знаний (на материале русских пословиц) // НТИ. Сер. 2. Информационные процессы и системы. № 1. М.: ВИНИТИ, 2011.

2 См.: *Малкова А.С.* Представление знаний в ценностных суждениях (на материале русских пословиц) // Двенадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2010 (20–24 сентября 2010 г., г. Тверь, Россия): Труды конференции. Т. 1. М.: Физматлит, 2010. С. 353–361.

3 См.: *Малкова А.С., Январев В.И.* Формальная модель семантики художественного текста (на материале русских пословиц) // Моделирование и анализ информационных систем. Т. 14. № 4. Ярославль: ЯрГУ, 2007. С. 43–53.

4 См.: *Фрейденберг О.М.* Поэтика сюжета и жанра / Ред. и коммент. Н.В. Брагинской. М.: Лабиринт, 1997.

5 См.: *Леви-Стросс К.* Мифологики: В 4 т. М.: Университетская книга, 1999.

6 См.: *Мелетинский Е.М.* Поэтика мифа. М.: Издат. фирма «Восточная литература» РАН, Школа «Языки русской культуры», 1995.

7 См.: *Лакофф Дж, Джонсон М.* Метафоры, которыми мы живем. М.: Едиториал УРСС, 2004.

8 См.: *Пермяков Г.Л.* Пословицы и поговорки народов Востока: Систематизированное собрание изречений двухсот народов. М.: Лабиринт, 2001.

9 См.: *Левин Ю.И.* Провербиальное пространство // Паремиологические исследования / Под ред. Г.Л. Пермякова. М.: Наука, 1984.

[10] См.: *Даль В.* Пословицы русского народа. Т. I–II. М., 1984.

[11] См.: *Потебня А.А.* Из лекций по теории словесности. Басня. Пословица. Поговорка // Потебня А.А. Теоретическая поэтика / Сост. А.Б. Муратов. М.: Высшая школа, 1990.

[12] См.: *Малкова А.С.* Разработка представления семантики ценностно-ориентированных текстов в базе знаний (на материале русских пословиц).

[13] См.: Словарь русских пословиц / В.М. Мокиенко, Ю.А. Ермолаева, А.А. Зайнульдинов и др.; Под ред. В.М. Мокиенко. М.: Астрель: АСТ, 2007.

[14] См.: *Малкова А.С.* Разработка представления семантики ценностно-ориентированных текстов в базе знаний (на материале русских пословиц).

[15] См.: *Малкова А.С.* Представление знаний в ценностных суждениях (на материале русских пословиц).

[16] См.: *Малкова А.С., Январев В.И.* Формальная модель семантики художественного текста (на материале русских пословиц).

Ю.К. Сергеев

АНАЛИЗ УГРОЗ БЕЗОПАСНОСТИ ВИРТУАЛЬНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

В данной статье приведены основные уязвимости систем виртуализации с учетом так называемых естественных уязвимостей, возникающих в результате ошибок программистов, воплотивших то или иное средство виртуализации. Приводится взаимоувязка уязвимостей с возможными путями реализации атак и необходимыми для этого условиями, что в конечном счете порождает угрозы безопасности информации, обрабатываемой в виртуальных информационных системах.

Ключевые слова: виртуализация, защита информации, угрозы, уязвимости, нарушитель, атаки.

На сегодняшний день мы наблюдаем активное повышение интереса к вопросу технологий виртуализации в современном мире[1]. И это не случайно, поскольку создание и функционирование столь популярных сегодня центров обработки данных немыслимо без их использования. При этом большинство компаний, вне зависимости от их размера, уже сегодня могут оценить экономический эффект от внедрения виртуальных информационных систем. Тем не менее подход к защите виртуальных информационных систем не то чтобы не систематизирован, но и не учитывает характерных для таких систем угроз.

Возникновение угроз, свойственных средам виртуализации

Основное отличие виртуальных систем от привычных физических заключается в необходимости изоляции ресурсов друг от

друга. И хотя эта проблема давно известна, в разрезе особенностей применения средств виртуализации она обретает новый облик. Ярким примером тому являются «облачные» технологии[2]. Рассмотрим ситуацию, в которой компания А организует cloud-сервис с использованием виртуализации как дополнительного уровня абстракции в целях распределения нагрузки на аппаратные ресурсы, представленные в виде десятков серверов. Могут ли в данном случае получатели услуги – некие компании Б и В – быть уверены в том, что их данные, которые обрабатываются на одном и том же физическом сервере, но в разных виртуальных машинах, действительно защищены?

Виртуализация позволяет разделять между компьютерами, входящими в состав различных информационных систем, следующие основные ресурсы:

– процессор;
– оперативную память;
– постоянную память;
– сетевые адаптеры.

Естественным является то, что зачастую информационные системы призваны обеспечивать обработку информации разных категорий и в разных целях. Благодаря технологиям виртуализации эксплуатация этих систем становится более удобной и гибкой. Однако при создании виртуальной инфраструктуры забываются либо не учитываются вопросы ее безопасности.

Инфраструктура, опирающаяся на технологию виртуализации, приобретает звенья, компрометация которых приводит к возможности несанкционированного доступа ко всем виртуальным компьютерам. Такими звеньями становятся гипервизор и средство управления виртуальной средой.

Используя гипервизор, злоумышленник может проводить любые действия в виртуальной инфраструктуре, причем полностью скрывать свои действия, так как информация может быть изменена путем выполнения кода вне оперативной памяти виртуальной машины, где действия злоумышленника могли бы быть зафиксированы. Атака на сервер управления приводит к тому, что злоумышленник может получить доступ к виртуальным машинам аналогичный физическому. В этом случае организационные меры по контролю доступа в серверные помещения уже не помогают. Загрузка компьютера – эта та стадия, на которой еще не работают средства защиты, а нарушитель уже может действовать. Таким образом, ключевыми элементами виртуальной инфраструктуры с точки зрения защиты информации становятся гипервизор и средство управления виртуальной инфраструктурой, которое зачастую является доверенным по отношению к гипервизору.

На текущий момент в открытом доступе не так много эксплойтов, которые бы позволяли выходить за пределы памяти виртуальной машины и тем более получать почти неограниченный доступ над всеми виртуальными машинами, но, учитывая рост применения технологии виртуализации, интерес к созданию атакующих средств в будущем злоумышленники будут проявлять больший. Однако такие широкие возможности доступны не только нарушителю, но и защищающимся, поэтому необходимо разрабатывать специализированные средства защиты, которыми можно было бы пользоваться.

Результат попытки систематизировать и свести воедино перечень угроз, соотнесенный с причинами их возникновения, на основе опыта, полученного при создании систем обеспечения безопасности информации различных организаций во многих отраслях деятельности, представлен в таблице.

Таблица

Соотнесение угроз виртуальных информационных систем и причин их возникновения

Угроза	Описание уязвимости	Путь реализации угрозы	Условия возникновения
Несанкционированный доступ к данным на диске другой виртуальной машины	Отсутствие механизмов контроля доступа Ошибка в ПО виртуализации Ошибка конфигурирования	Подключение диска к виртуальной машине злоумышленника Копирование виртуальной машины со всеми данными	Даже когда разграничительная система защиты настроена внутри операционной системы, СУБД и приложений, виртуальные диски могут быть доступны для подключения (в некоторых случаях только на чтение). Отсутствие контроля пользователей в части контроля доступа к дискам виртуальных машин может привести к копированию виртуальной машины целиком вместе с конфиденциальной информацией на незащищенные носители, где к защищаемой информации можно получить доступ в обход средств защиты от несанкционированного доступа

Продолжение таблицы

Угроза	Описание уязвимости	Путь реализации угрозы	Условия возникновения
Перехват информации, передаваемой по сети между виртуальными машинами	Отсутствие средств фильтрации трафика между виртуальными машинами Прослушивание информации на виртуальном коммутаторе Ошибка в ПО виртуализации Ошибка конфигурирования	Перевод виртуального адаптера в promiscuous-режим Прямой доступ к виртуальным машинам в обход внешних средств межсетевого экранирования	Сетевая инфраструктура часто базируется на основе программно-аппаратных средств контроля межсетевого трафика, использование которых осложнено в виртуальной инфраструктуре
	Единое время и зависимость от одной аппаратной платформы	Атака типа «человек посередине», например TCP hijacking	Так как создание TCP ISN (initial sequence number) зависит от функции генерации псевдослучайных чисел, то атакующему на виртуальную машину из виртуальной среды гораздо проще подобрать TCP ISN и реализовать TCP hijacking атаку
Отказ в обслуживании в результате захвата ресурсов	Ошибка конфигурирования Отсутствие механизмов квотирования ресурсов[3]	Загрузка центрального процессора виртуальной машины Отъем оперативной памяти Активное использование	Создание виртуальных информационных систем производится без учета такого свойства безопасности информации, как доступность. Часто пользователь информационных систем не формирует бизнес-требований

Продолжение таблицы

Угроза	Описание уязвимости	Путь реализации угрозы	Условия возникновения
		виртуальных жестких дисков	по приоритезации доступности тех или иных информационных систем
Перехват данных во время миграции виртуальной машины	Отсутствие механизмов защиты передачи трафика	Прослушивание трафика	Использование общей ЛВС для миграции. Память виртуальной машины синхронизируется через общую сеть
Несанкционированный доступ к данным в памяти другой виртуальной машины	Отсутствие механизмов контроля доступа	Использование встроенных возможностей доступа к памяти других виртуальных машин (backend/frontend-драйвера, общие области памяти)	Информационные системы, обрабатывающие информацию разных категорий, размещаются в одной виртуальной среде. При этом не предусматривается использование механизмов разграничения доступа к памяти
	Ошибка в ПО виртуализации	Эксплуатация классических уязвимостей (переполнение буфера, уязвимость форматной строки и т. п.) с последующим доступом к оперативной памяти атакуемой виртуальной машины	
	Ошибка конфигурирования	Неверная настройка прав разграничения доступа к памяти другой виртуальной машины	

Продолжение таблицы

Угроза	Описание уязвимости	Путь реализации угрозы	Условия возникновения
Несанкционированный доступ к данным в результате компрометации сервера управления	Отсутствие сетевого контроля доступа к серверу управления Отсутствие механизмов контроля доступа Ошибка конфигурирования Ошибка в ПО виртуализации	Атака на сервер управления	Атака на сервер управления позволяет злоумышленнику получить интерфейс для обхода средств защиты, установленных на виртуальной машине, в процессе ее загрузки
Отказ в обслуживании за счет роста «динамических» дисков	Ошибка конфигурирования Отсутствие механизмов предупреждения[4]	Создание виртуальных машин с «динамическими» дисками и одновременное заполнение их файловых систем данными	Разрешение использования в промышленной системе динамических дисков и отсутствие контроля за их использованием, а также планирования дискового пространства
Несанкционированный доступ к данным в результате компрометации SAN	Ошибка конфигурирования	Монтирование LUN с виртуальными машинами в другие системы	При проектировании SAN не учитываются требования по безопасности с учетом обработки защищаемой информации разных категорий в виртуальных информационных системах
Компрометация ключевой информации,	Функция сохранения состояний (snapshot)	Сохранение snapshot с оперативной памятью на диск	Ключи шифрования, загруженные в оперативную память даже из защищенного носителя, могут быть извле-

Продолжение таблицы

Угроза	Описание уязвимости	Путь реализации угрозы	Условия возникновения
используемой в криптографических операциях			чены из snapshot либо получены путем прямого доступа к оперативной памяти
	Отсутствие механизмов контроля доступа	Прямой доступ к ключам в памяти другой виртуальной машины	
	Зависимость от времени и аппаратной платформы при генерации ключевой информации	Replay-атака на One-Time-аутентификацию	Прослушав одноразовый пароль в сети, можно откатить из snapshot виртуальную машину назад во времени, после чего она примет прослушанный ранее пароль как верный
		Атака на поточные шифры	Существует вероятность того, что две виртуальные машины (например, копии одной и той же), использующие программный генератор псевдослучайных чисел, будут шифровать поток данных на одном и том же ключе. В этом случае расшифровать их можно после применения простого XOR
		Подбор ключей	Так как генерация ключевой информации полагается на одну и ту же программно-аппаратную платформу, повышается вероятность успешной криптографической атаки при высоком уровне подготовки атакующего

Продолжение таблицы

Угроза	Описание уязвимости	Путь реализации угрозы	Условия возникновения
		Атаки на время-зависимые протоколы Ошибка конфигурирования	За счет отката времени с использованием snapshot существует вероятность реализации атаки на различные протоколы, зависимые от времени либо базирующиеся на истории выполнения: американский стандарт генерации цифровой подписи DSS[5], российский ГОСТ Р 34.10-2001[6], механизм Perfect Forward Secrecy (PFS)[7] в технологии SSL или IPSec
Деактивация средств защиты	Отсутствие механизмов контроля доступа Ошибка в ПО виртуализации	Изменение данных в оперативной памяти Изменение файлов конфигураций	Возможность осуществления чтения и записи оперативной памяти виртуальных машин из других приводит к тому, что пользователь, не обладающий правами администратора в защищаемой информационной системе, имеет возможность деактивировать средства защиты без его ведома
Отказ в обслуживании сети из за «принудительной» миграции	Ошибка конфигурирования	Навязывание постоянной миграции виртуальных машин между узлами кластера среды виртуализации	В результате ошибок конфигурирования пользователи могут вызвать отказ в обслуживании в результате многократной миграции виртуальной машины с одного хоста на другой и появления задержки в сети, соединяющей кластер

Окончание таблицы

Угроза	Описание уязвимости	Путь реализации угрозы	Условия возникновения
Неотслеживаемость действий злоумышленника	Функция сохранения состояний (snapshot) Отсутствие механизмов контроля доступа	Сохранение состояния – осуществление несанкционированных действий – откат предыдущего состояния	Использование snapshot приводит к тому, что регистрация событий средствами «классических» средств защиты недостаточна, так как пользователи, имеющие права создания копии состояния, всегда могут откатиться в него, осуществив несанкционированные действия по отношению к данным, обрабатываемым в виртуальной информационной системе
Несанкционированный доступ к остаточной информации	Функция расширения «динамических» дисков Функция динамического выделения оперативной памяти	Создание и запуск злоумышленником новой виртуальной машины	В угоду производительности средство виртуализации не обеспечивает очистки памяти перед ее выделением, полагаясь на выполнение данной операции средствами операционной системы в виртуальной машине

Приведенная таблица не включает в себя информацию, необходимую для ранжирования угроз по степени их важности, и не определяет конкретные средства защиты, которые могут помочь снизить их возникновение. Часть угроз может быть минимизирована за счет встроенных механизмов систем виртуализации, другая же требует применения новых специализированных средств защиты.

Тем не менее указанный перечень позволяет сформировать общие требования по защите виртуальных информационных систем.

Требования по защите виртуальных информационных систем

Состав перечня требований по защите виртуальных информационных систем зависит от актуальности той или иной угрозы и предполагает, что стоимость защиты от угрозы не превышает ценность самой защищаемой информации, за исключением категорий информации, защита которой осуществляется в соответствии с положениями законодательства РФ, включающих обязательный набор требований по защите.

По результатам анализа угроз, представленных в таблице, можно выделить следующие требования по защите виртуальных информационных систем, которые рекомендуется корректировать в зависимости от ситуации:

1) проектирование информационных систем, реализуемых в виртуальной среде, должно проводиться с учетом требований по безопасности обрабатываемой информации;

2) в промышленной системе должно быть запрещено использование функции создания snapshot. Допускается их использование только в тестовой среде или среде для разработчиков, в которых не ведется обработка конфиденциальной информации;

3) должны быть предусмотрены механизмы разграничения доступа к оперативной памяти, виртуальным жестким дискам (дискреционный или полномочный доступ в зависимости от категории информации);

4) должно быть обеспечено журналирование действий пользователей средства управления виртуальной инфраструктурой;

5) должны использоваться специализированные средства защиты виртуальных машин. Указанные средства не просто должны выполняться в виртуальной среде, но и использовать преимущества, получаемые от возможности использования функций гипервизора для защиты виртуальных машин;

6) должны использоваться механизмы квотирования ресурсов (особенно для виртуальных сред, в которых обрабатывается конфиденциальная информация нескольких компаний);

7) необходимо производить генерацию ключевой информации с использованием аппаратных датчиков случайных чисел либо генерировать ключи вне виртуальной среды;

8) необходимо предусматривать механизмы очистки остаточной информации.

Заключение

В результате проведенного анализа был сформирован перечень возможных угроз для виртуальных информационных систем, которые должны быть дополнительно рассмотрены с целью определения степени их опасности и вероятности возникновения, а также предложен набор общих требований по защите виртуальных информационных систем, которые могут быть использованы на стадии разработки технического задания на создание системы защиты.

На основе полученных результатов можно выделить основные направления защиты виртуальных информационных систем, которые необходимо развивать для минимизации обозначенных угроз. Последующие исследования могут быть направлены на классификацию угроз по степени ущерба, разработки основных механизмов защиты, учитывающих характерные угрозы виртуальных информационных систем.

Примечания

1 См.: *Rutkowska J.* Security Challenges in Virtualized Environments // RSA Conference. San Francisco, 2008.

2 См.: *Aviram A., Hu S., Ford B., Gummadi R.* Determinating timing channels in compute clouds // Proceedings of the 2010 ACM Workshop on Cloud Computing Security. Chicago, Illinois, USA, October 8, 2010.

3 Большинство средств виртуализации, предназначенных для промышленной эксплуатации, имеют данный функционал.

4 Большинство средств виртуализации, предназначенных для промышленной эксплуатации, имеют данный функционал.

5 Для генерации простых чисел используются SHA и начальное число SEED. Как раз SEED и может быть повторен из-за восстановления виртуальной машины из snapshot.

6 Хотя ГОСТ Р 34.10-2001 и не описывает процедуры генерации простых чисел, принципиально условия появления уязвимости не отличаются от DSS в случае, если зависят от программного датчика генерации псевдослучайных чисел.

7 При использовании PFS злоумышленник, получивший сессионный ключ, не может скомпрометировать предыдущие сессионные ключи за счет применения «соли», генерируемой псевдослучайным образом, которая также может быть предугадана из-за восстановления виртуальной машины из snapshot.

Р.Р. Гилязов

ОЦЕНКА ВРЕМЕНИ, ПРОШЕДШЕГО МЕЖДУ ДВУМЯ СОБЫТИЯМИ, В ОПЕРАЦИОННОЙ СИСТЕМЕ

В ходе данной работы будут рассмотрены практические аспекты и обозначена проблематика задачи измерения времени, прошедшего между двумя событиями. Также в соответствии с разработанной методикой будет проведена серия практических экспериментов, которые устанавливают влияние операционной системы, время реакции защитного программного обеспечения на вредоносное воздействие, а также оценивают некоторые свойства эвристики для конкретно выбранного программного продукта.

Ключевые слова: измерение времени, информационная безопасность, влияние операционной системы, эвристика, время реагирования защитного ПО, временные датчики.

В операционной системе в произвольные моменты времени в процессе функционирования происходят различного рода события. Представляет интерес задача подсчета времени, прошедшего между двумя событиями, особенно если одно из них влечет другое. В класс таких задач попадает множество тем, например оценка времени реакции защитного программного обеспечения на вредоносное воздействие, влияние операционной системы на время работы некоторых последовательностей машинных инструкций, профилирование и т. п. Однако по причине воздействия внешних факторов подсчет времени является нетривиальной задачей.

Постановка задачи

– Анализ средств и методик, позволяющих оценить количество времени, прошедшего между двумя событиями в операционной системе.

– Обозначение проблематики оценки количества времени, прошедшего между различными событиями.

– Проведение экспериментов, позволяющих оценить степень воздействия операционной системы на исполняемый код.

– Проведение экспериментов, позволяющих оценить время реакции защитного программного обеспечения на вредоносное воздействие.

Определение события
Пример 1

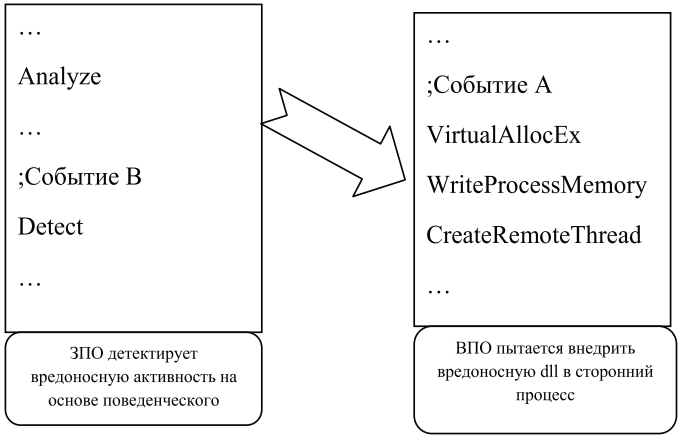
```
mov eax, 0 // событие A

;//
;// последовательность машинных инструкций M
;//

cmp eax, 1 // событие B
```

Расстояние между событиями A и B в данном примере – это время работы последовательности машинных инструкций M.

Пример 2



Расстояние между событиями A и B в данном примере – реакция защитного программного обеспечения (ЗПО) на вредоносное воздействие. В данном случае вредоносное программное обеспечение (ВПО) пытается внедрить dll в сторонний процесс, для получения доступа к его закрытому адресному пространству.

Введение функции расстояния между событиями

Для оценки расстояния между двумя событиями A и B введем функцию ρ(A, B).

Проанализируем аппаратные компоненты современного компьютера, которые можно использовать для введения такой функции[1,2].

1. *Programmable Interval Timer (PIT) (8253/8254)*

Появился у первых ПК фирмы IBM. Имеет три эквивалентных канала. На вход каждого канала подается тактовая частота синхронизации – 1,193 либо 14,31818 МГц, точность порядка – 1 мс. Стандартная продолжительность одного тика таймера в Windows – 10 мс. Каждые 10 мс таймер генерирует прерывание, и соответствующий обработчик увеличивает системное время на 10 мс. Точность измерений сильно падает, если по какой-либо причине обработчик не сработает после соответствующего прерывания, что вполне вероятно. Windows использует PIT-аймер в основном для планирования потоков, для измерения времени ОС чаще использует другие, более точные таймеры, переход на PIT осуществляется только тогда, когда дополнительные таймеры недоступны.

2. *Real Time Clock (RTC).*

Общее название класса микросхем. Появился впервые в IBM-AT элемент (MC146818), дополнительно к 8254. Так же, как и с 8254, доступ обеспечивается через порт ввода-вывода. Точность часов – 1 мс, стандартная частота работы – 32768 Гц. RTC-таймер может быть перепрограммирован, в результате чего часы будут идти медленнее или быстрее. Точность показаний зависит как от состояния питающей батарейки (литиевый аккумулятор), так и от качества реализации микросхемы RTC со всеми обслуживающими ее компонентами. Некоторые из реализаций обнаруживают значительную нестабильность на временных интервалах порядка десятых долей секунды, что уменьшает точность измерений.

3. *Таймер APIC*

Обладает большинством свойств RTC. Создавался для синхронизации процессоров в многопроцессорных системах. Имеет низкую точность и невысокую стабильность работы ввиду конструктивных

недоработок. Существенный недостаток также в том, что некоторые реализации ведут себя нестабильно при определенных настройках электропитания. Присутствует не во всех современных системах. Стандартная частота таймера – 18,2 Гц, стандартное время тика – 10 мс. APIC-ядра используют APIC-таймер в качестве основного таймера системы, при этом величина одного тика составляет 15 мс.

4. *Таймер PM (ACPI-таймер)*

По умолчанию тактируется частой 3,579545 MHz (тактовая частота PIT, разделенная на четыре), что обеспечивает точность измерений порядка 0,3 мс. ACPI-ядра используют PM-таймер в качестве основного таймера системы. Некоторые APCI-контроллеры динамически изменяют частоту процессора или усыпляют его в паузах между работой для лучшего охлаждения, поэтому может присутствовать высокая корреляция между таймером PM и счетчиком TSC.

5. *Таймер HPET*[3] *(High Precision Event Timers)*

Тактируется частотой от 10 MHz, при которой время одного тика составляет от 0,1 мс при точности порядка ±0,2% на интервалах от 1 до 100 мс. HPET планировался как замена программируемому интервальному таймеру PIT и возможностям RTC по генерации прерываний. В сравнении с прочими таймерами HPET имеет более высокую разрешающую и большую интервальную способность (число, по которому срабатывает таймер, хранится в 64-битном счетчике).

Примечание: 8254 и RTC способны аналогично HPET работать в режиме единичного срабатывания, однако процесс их настройки столь медленен, что их не используют в областях, требующих высокой разрешающей способности счетчика; вместо этого RTC/8254 обычно используются в периодическом режиме с малыми интервалами (порядка нескольких миллисекунд) с пропуском нужного количества интервалов. Подобное приводит к появлению частых прерываний, даже если они не нужны программе. При использовании HPET дополнительные прерывания не нужны, так как настройка HPET для единичного срабатывания существенно проще (и требует меньшего времени), чем для RTC/8254.

6. *Счетчик TSC (Time Stamp Counter)*

Увеличивается на 1 каждый такт работы процессора с момента его включения. Для чтения используется команда RDTSC. В современных процессорах не зависит от технологии энергосбережения.

Если в качестве функции ρ(A, B) выбирать таймер, то возникают проблемы следующего плана:

– сильная зависимость от оборудования, и особенно от производительности системы;
– низкая точность.

Такие недостатки меньше всего проявляются в счетчике TSC. В дальнейшем положим, что ρ(A, B) = RDTSC(B) − RDTSC(A). Данная функция должна быть строго положительной, поэтому событию B должно обязательно предшествовать событие A и данные события должны быть различны. Причем значение функции не определено, если одно из событий, являющихся аргументами данной функции, не произошло.

Рассмотрим практически аспекты применения данной функции при проведении экспериментов.

Практические аспекты применения функции ρ(A, B) для измерений

Для измерения расстояния между двумя событиями была выбрана функция ρ(A, B) = RDTSC(A) − RDTSC(B). Рассмотрим практические аспекты ее применения. Современный процессор имеет сложную архитектуру, которую необходимо учитывать при проведении измерений. Пусть A и B события. Для ускорения вычислений в процессоре присутствует конвейер. Если машинные инструкции, находящиеся между A и B, по каким-то причинам простаивают, процессор для оптимизации может выполнить другие команды, которые не попадают в измеряемый диапазон, что исказит результат. Чтобы избежать этого, необходимо использовать команды приоритетного выполнения, очищающие конвейер[4]. Одной из таких команд является CPUID. Важно помнить, что входной параметр этой команды при проведении измерений должен иметь одно и то же значение (передается через регистр eax).

Пример использования RDTSC

```
XOR EAX,EAX
CPUID
RDTSC
MOV[var],EAX;
; // событие A
;//. . .
;// событие B
XOR EAX,EAX CPUID RDTSC
SUB EAX, [var]
```

Конвейер процессора характеризуется пропускной способностью и латентностью. Если ρ(A, B) < латентности, то результаты

измерений крайне неточны. Также конвейер характеризуется длиной очевидно, что если p(A, B) < длины конвейера, то результаты измерений некорректны[5].

Так как операционная система поддерживает многозадачность, то если p(A, B) > кванта переключения контекста, точность измерений падает из-за смены контекста процесса.

Кроме того, общая аппаратная конфигурация должна быть пригодна для измерений. Как правило, аппаратная погрешность незначительна и не превышает 1–3%, однако на некоторых тестируемых системах конфигурация вносила погрешность до 100%.

Если проводятся серии последовательных измерений, искажения вносят различные механизмы, разработанные для оптимизации, в основном это кэш и буферы ввода / вывода[6], однако эти механизмы при необходимости можно отключить.

Так как измерения проводятся в среде операционной системы, она оказывает непосредственное влияние на проводимые эксперименты. Ставились эксперименты, в которых операционная система настолько влияла на проводимые измерения, что они отличались на несколько порядков.

При определенных экспериментах методика измерений очень сильно влияла на сами измерения, что не позволяло получать хоть сколько-нибудь значимые оценки.

Проведение экспериментов

Испытательный стенд

ЭВМ

Для получения более точных оценок используется однопроцессорная система.

Основные характеристики:

ЦП: Pentium 4 2.8Ghz, HyperThreading отключен, FSB 200 MHz
ОЗУ: 2048 мб, DDR3 6400

ЖД: 120 gb, 7200 rpm

Операционная система

Для того чтобы уменьшить воздействие посторонних факторов на проводимые эксперименты, на испытательный стенд установлена оригинальная версия операционной системы с минимально необходимым для конкретного испытания программным обеспечением.

Защитное программное обеспечение

Общедоступное коммерческое защитное программное обеспечение. Предполагается, что все настройки являются настройками по умолчанию.

Компоненты, разработанные для тестирования

1. Был реализован компонент, который работает вне операционной системы и позволяет получить монопольный доступ к ресурсам компьютера.

2. Были разработаны компоненты, которые распознаются защитным ПО как вредоносные.

3. Был разработан компонент, при помощи которого отслеживается, сколько машинных тактов проходит между двумя событиями в операционной системе.

Методика подсчета метрик

Аспекты проведения экспериментов были проанализированы в теоретической части. Для оценки расстояния между событиями используется команда RDTSC.

Серия испытаний № 1

Цель: Выявление влияния операционной системы на выполнение последовательных инструкций в одном адресном пространстве одного процесса.

Методика эксперимента

Интересной задачей является установление влияния ОС на выполнение промежутков кода различного содержания. Для данного эксперимента был написан компонент, который работает вне операционной системы и позволяет получить монопольный доступ к ресурсам операционной системы. Был переписан MBR (Master Boot Record) загрузочного диска таким образом, чтобы управление отдавалось сначала на наше приложение, затем на загрузчик операционной системы. Конечно, можно выполнить вычисления для заданного промежутка кода вручную, однако такая методика не учитывала бы влияние аппаратной составляющей. Очевидно, что для корректности экспериментов требуется аккуратным образом выбирать промежутки кода. В теоретической части работы указаны проблемы вычисления расстояния между событиями и даны некоторые указания для проведения серий испытаний. Выберем участки кода различных типов и подсчитаем время их работы в режиме монопольного доступа к оборудованию и сравним с временем работы в среде операционной системы.

Описание экспериментов

1. Работа только с регистрами без участия оперативной памяти.

2. Работа преимущественно с оперативной памятью. При работе с памятью отключено кэширование и буферизация.

3. Работа преимущественно со стеком.

Количество прогонов в каждом эксперименте было равно 10^6. Для усреднения бралось среднее арифметическое.

Регистры	Стек	Память	φ
ДА	НЕТ	НЕТ	1
ДА	НЕТ	ДА	1,31
ДА	ДА	НЕТ	1,43
ДА	ДА	ДА	1,36

Примечание. φ = ρw / ρe, где ρw – время работы промежутка кода в среде операционной системы Windows, ρe – время работы промежутка кода в режиме эксклюзивного доступа к оборудованию.

Полученные результаты

В ходе проведения данных серий первого эксперимента существенного влияния операционной системы не было установлено. Для остальных экспериментов были обнаружены воздействия, обусловленные, по всей видимости, механизмами кэширования стека и трансляции виртуальных адресов в физические. Полноценно природа влияния операционной системы изучена не была, так как это выходит за рамки данной работы.

Серия испытаний № 2

Цель: Выявление времени реакции защитного программного обеспечения на вредоносное воздействие.

Методика эксперимента

Интересной задачей является установление времени реакции на вредоносное ПО (ВПО) защитным программным обеспечением (ЗПО). В качестве вредоносного ПО были написаны небольшие примитивы, которые однозначно детектируются ЗПО. Дополнительно для данного эксперимента был написан компонент, который позволяет вычислять количество машинных тактов, прошедших между событием, произошедшим в адресном пространстве вредоносного примитива, и событием, являющимся реакцией на него ЗПО. Очевидно, что для корректности экспериментов требуется правильно разработать такую утилиту для ВПО. Так как мы обладаем исходными кодами, получать значение TSC несложно, но получение этого значения во время реакции ЗПО сопряжено с некоторыми трудностями. В данной серии испытаний под реакцией ЗПО будет считаться момент возникновения информационного окна с сообщением о вредоносной активности.

Описание экспериментов

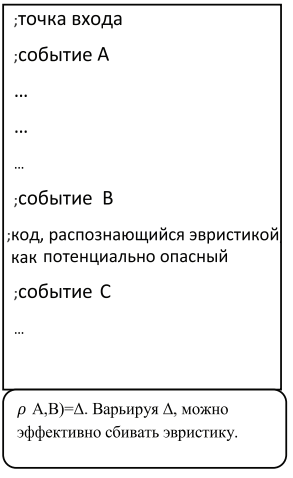
1. Вредоносный файл находится на флэш-носителе. Он содержит сигнатуру, известную защитному ПО. Измеряется количество

машинных тактов, прошедших с момента включения режима сканирования до момента обнаружения вредоносного файла.

2. Вредоносный файл находится на флэш-носителе. Пользователь активирует его. Сигнатура известна ЗПО.

3. Вредоносный файл находится на флэш-носителе. Пользователь активирует его. Сигнатура неизвестна ЗПО. Однако ЗПО способно обнаружить ВПО, используя механизмы эвристического анализа.

4. Вредоносный файл находится на флэш-носителе. Пользователь активирует его. Сигнатура неизвестна ЗПО. ЗПО способно обнаружить ВПО, используя механизмы эвристического анализа. Однако перед промежутком кода, который распознается эвристикой как вредоносный, стоит значительное количество машинных инструкций, которые не распознаются эвристикой как потенциально опасные (см. рисунок).



Полученные результаты

В ходе серий испытаний были получены конкретные значения для экспериментов № 1 и 2, которые являются статистически устойчивыми. Для эксперимента № 3 не удалось получить конкретных устойчивых значений.

Время работы	Среднеквадратичное отклонение	Номер эксперимента
316418419	22430711	1
2744532904	424891383	2
4708124743	3255607282	3

Для эксперимента № 4 удалось выбрать минимальную Δ так, чтобы эвристический анализатор не обнаруживал вредоносной активности.

Заключение

В ходе проведенных исследований была обозначена проблематика оценки времени, прошедшего между двумя событиями в системе. Были выявлены особенности, влияющие на точность оценок и корректность испытаний. Были поставлены серии практических экспериментов, которые выявляют влияние операционной системы на время выполнения кода. Также частично была рассмотрена задача оценки времени реагирования защитного программного обеспечения на вредоносное воздействие. В некоторых экспериментах для этой задачи установлена крайняя нестабильность получаемых результатов, что приводит к трудностям построения математической статистики и проведения комплексного анализа. Дополнительно были произведены некоторые исследования в области эвристики защитного программного обеспечения и было построено ВПО, не обнаруживаемое эвристическими механизмами.

Примечания

1 См.: *Касперски К.* Разгон и торможение Windows NT // Системный администратор. 2004. № 8.

2 См.: *Руссинович М., Соломон Д.* Внутреннее устройство Microsoft Windows: Windows Server 2003, Windows XP и Window 2000: Пер. с англ. 4-е изд. СПб.: Питер, 2008.

3 См.: Intel Corporation (October 2004), IA-PC HPET (High Precision Event Timers) Specification (revision 1.0a) [Электронный ресурс]. [USA, 2004]. URL: http://www.intel.com/hardwaredesign/hpetspec_1.pdf (дата обращения: 20.12.2010).

[4] См.: Intel, Using the RDTSC Instruction for Performance Monitoring [Элекронный ресурс]. [USA: Intel Corporation, 1997]. URL: http://pasta.east.isi.edu/algorithms/IntegerMath/Timers/rdtscpm1.pdf (дата обращения: 20.12.2010).

[5] См.: *Касперски К.* Техника оптимизации программ. Эффективное использование памяти. М.: БХВ-Петербург, 2003.

[6] См.: *Руссинович М., Соломон Д.* Указ. соч.

В.В. Черняковский

СПОСОБ ДИНАМИЧЕСКОГО ПОИСКА ГЛОБАЛЬНЫХ ПЕРЕМЕННЫХ ЯДРА В ОПЕРАЦИОННЫХ СИСТЕМАХ СЕМЕЙСТВА WINDOWS NT

В данной работе представлен новый способ динамического поиска глобальных переменных ядра, в ней приводятся теоретическое обоснование возможности его создания, а также экспериментальное обоснование возможности его практического использования.

Ключевые слова: операционная система, семейство Windows NT, глобальные переменные ядра, отладчик ядра, драйвер, руткит, программные средства защиты.

Введение

Со стремительным развитием информационных технологий средства современной вычислительной техники становятся неотъемлемой частью при организации производства и построении бизнес-процессов на предприятиях разной величины. Возрастающее количество используемых аппаратных средств и увеличение функциональных возможностей создаваемых программных средств усложняют процедуры контроля происходящих в информационных системах процессов, что приводит к снижению их защищенности, а также отказоустойчивости. Увеличение функциональных возможностей программных средств неизбежно ведет к появлению недекларируемых возможностей программного обеспечения и ошибок в логике их работы, информация о которых может быть применена для подготовки и проведения атак на информационные системы[1].

В последнее время стремительно стали развиваться и распространяться новые виды вредоносных программ (malware), разрабатываемых с применением механизмов руткит-технологий, которые позволяют нарушителю безопасности скрывать факты своего при-

сутствия и активной деятельности в пораженной системе. Такие вредоносные программы носят название руткитов[2] (rootkit), они представляют огромную угрозу для безопасности информационных систем, поскольку их обнаружение является проблемой для современных средств защиты, ориентированных на обнаружение и противодействие другого рода угрозам.

Для противодействия такого рода угрозам разрабатывается отдельный класс средств защиты, известных как системы предотвращения вторжений (Host Intrusion Prevention System – HIPS). Эти программные средства защиты основаны, как правило, на использовании штатных механизмов защиты, предоставляемых самой операционной системой (ОС). В свою очередь штатные механизмы защиты ОС представляют собой некоторые алгоритмы управления ключевыми компонентами ядра ОС (диспетчер системных сервисов, диспетчер процессов и потоков, диспетчер ввода / вывода и др.). Каждый из таких ключевых компонентов ОС представляет собой некоторый объект ядра, который описывается определенной структурой в памяти. Чтобы иметь возможность управления (создания, удаления, изменения) и контроля над этими объектами, ОС хранит их адреса в глобальных переменных ядра.

Поиск адресов таких глобальных переменных ядра, с помощью которых осуществляется доступ к ключевым объектам ядра ОС, является первостепенной и одной из важнейших задач, стоящих перед разработчиками программных средств защиты. Поэтому разработка новых и совершенствование существующих способов нахождения глобальных переменных ядра ОС являются весьма существенными задачами в сфере защиты информационных систем. Особенно актуальны эти задачи для современных ОС семейства Windows NT, которая на данный момент является самой распространенной платформой для корпоративных решений, используемых различными организациями в своих распределенных информационных системах.

Целью данной работы является разработка нового способа нахождения адресов глобальных переменных ядра, применимого в ряде современных ОС семейства Windows NT.

Для достижения поставленной цели необходимо решение следующих задач:

- теоретически обосновать возможность создания нового способа поиска глобальных переменных ядра в ОС семейства Windows NT;
- экспериментально обосновать возможность практического использования разработанного способа.

При написании данной работы использовались литература и публикации по следующим тематикам:

– архитектура и внутреннее устройство ОС семейства Windows NT;
– руткит-технологии, основные механизмы и техники, способы противодействия;
– известные способы нахождения глобальных переменных ядра;
– разработка драйверов для современных ОС семейства Windows NT.

Теоретическое обоснование

Идея разработки нового способа динамического поиска глобальных переменных ядра основана на результатах исследований, опубликованных Эдгаром Барбосой[3]. В ходе исследования ядра ОС Windows XP (NT 5.1) им было обнаружено некоторое изменение в описании одной из ключевых структур ядра относительно ОС Windows 2000 (NT 5.0). Изменение заключается в том, что в структуре области управления процессором **KPCR** (Kernel Processor Control Region), которая описывает состояние каждого процессора в системе, поле **Reserved**[2] изменило свое название на **KdVersionBlock**. Помимо этого, вместо нулевого значения это поле содержало в качестве своего значения некоторый адрес, т. е. являлось указателем на некую внутреннюю структуру ядра. В ходе проведения дальнейших исследований ядра ОС Windows XP было выяснено, что поле **KdVersionBlock** структуры **KPCR** содержит указатель на структуру **nt!KdVersionBlock**, которая, в свою очередь, содержит указатель на известную структуру **nt!KdDebuggerDataBlock**. Эти внутренние структуры ядра ОС используются отладчиком ядра для быстрого определения текущего состояния ОС и ее основных компонентов, а также для предоставления необходимой отладочной информации, например о причинах краха системы. Полные описания этих структур (под именами **DBGKD_GET_VERSION** и **KDDEBUGGER_DATA** соответственно) можно найти в файле **wdbgexts.h**, входящем в состав заголовочных файлов набора для разработки драйверов (Windows Driver Kit – WDK), доступном на официальном сайте компании Microsoft Corporation.

Данные структуры в совокупности содержат список большинства глобальных переменных ядра, используемых различными компонентами ОС. Результаты анализа содержимого данных структур позволили сделать вывод о том, что в них находится необходимое и достаточное количество информации, позволяющее динамически определять текущее состояние ОС и ее основных компонентов. На

рис. 1 представлена схема, которая демонстрирует взаимосвязь между структурами, предназначенными для использования отладчика ядра.

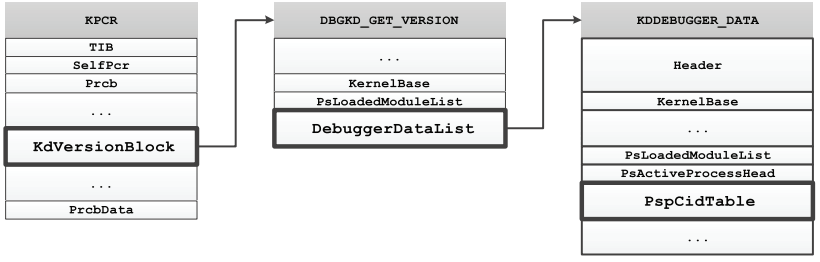


Рис. 1. Взаимосвязь структур, содержащих отладочную информацию

Используя полученную взаимосвязь, можно составить алгоритм нахождения адресов большинства глобальных переменных ядра. Этот алгоритм и положен в основу предлагаемого способа динамического поиска глобальных переменных ядра, среди которых наибольший интерес представляют следующие:

- **KernelBase**, адрес загрузки образа ядра ОС в памяти;
- **PsActiveProcessHead**, указатель на голову списка активных процессов в системе;
- **PsLoadedModuleList**, указатель на голову списка загруженных исполняемых модулей ядра (драйверов) в системе;
- **PspCidTable**, указатель на таблицу дескрипторов (описателей) существующих в системе процессов и потоков.

Алгоритм поиска некоторой глобальной переменной ядра состоит из следующих этапов:

1) получить адрес структуры **KPCR**, описывающей состояние текущего процессора (он хранится в регистре **fs** в 32-разрядных версиях ОС и регистре **gs** в 64-разрядных[4]);
2) получить адрес структуры **DBGKD_GET_VERSION** (он хранится в поле **KdVersionBlock** структуры **KPCR**);
3) получить адрес структуры **KDDEBUGGER_DATA** (он хранится в поле **DebuggerDataList** структуры **DBGKD_GET_VERSION**);
4) получить адрес интересующей глобальной переменной (значение соответствующего поля структуры **KDDEBUGGER_DATA**).

Использование предлагаемого способа динамического поиска глобальных переменных ядра позволяет избежать многих принципиальных проблем, встающих перед разработчиками современных средств защиты для ОС семейства Windows NT.

Поддержка различных версий ОС. Адреса глобальных переменных ядра отличаются в различных версиях ядра ОС (как в ядрах новых версий ОС, так и в ядрах, предназначенных для различных типов процессоров), даже очередной пакет обновлений (Service Pack) в рамках одной версии ОС может внести достаточные изменения в бинарный образ ядра, чтобы изменить адреса глобальных переменных. Помимо этого, в версиях ОС начиная с Windows Vista (NT 6.0) введен новый механизм защиты – рандомизация адресного пространства[5] (Address Space Layout Randomization – ASLR). Его суть заключается в том, что бинарные образы ядра, его модулей, динамических библиотек и приложений загружаются в память по вычисляемым случайным образом адресам, что нарушает работу не только вредоносных программ, но и программных средств защиты. Поэтому использование статически заданных адресов и смещений относительно других объектов ядра не является достаточно эффективным способом.

Производительность и стабильность. Многие средства защиты используют сигнатурный поиск для нахождения адресов глобальных переменных, что позволяет устранить проблему поддержки различных версий ОС. То есть происходит сканирование физической памяти на предмет совпадения содержимого ячеек памяти с некоторым заданным шаблоном (последовательностью байт). Такой способ заметно теряет свою эффективность на машинах с большим объемом физической памяти, тем более нет никаких гарантий, что найденное совпадение – это не случайный мусор в памяти, что может привести к краху всей системы. К тому же не каждая глобальная переменная ядра имеет известную сигнатуру, что является серьезным ограничением на использование сигнатурного поиска.

На основе вышеизложенного можно утверждать, что предлагаемый способ обладает рядом преимуществ перед другими известными способами поиска адресов глобальных переменных ядра:

- платформенная независимость;
- быстродействие и надежность;
- простота реализации и использования.

Экспериментальное обоснование

На основе вышеописанного способа был разработан модуль ядра для динамического получения адресов некоторых глобальных переменных ядра ОС семейства Windows NT.

На листинге 1 представлены описания необходимых структур на языке программирования C, который используется для разработки драйверов для ОС семейства Windows NT.

```
// Описание структуры ядра nt!KdVersionBlock
typedef struct _DBGKD_GET_VERSION
{
     USHORT                  MajorVersion;
     USHORT                  MinorVersion;
     UCHAR                   ProtocolVersion;
     UCHAR                   KdSecondaryVersion;
     USHORT                  Flags;
     USHORT                  MachineType;
     UCHAR                   MaxPacketType;
     UCHAR                   MaxStateChange;
     UCHAR                   MaxManipulate;
     UCHAR                   Simulation;
     USHORT                  Unused[1];
     ULONG64                 KernelBase;
     ULONG64                 PsLoadedModuleList;
     ULONG64                 DebuggerDataList;

} DBGKD_GET_VERSION, *PDBGKD_GET_VERSION;


// Описание структуры ядра nt!KdDebuggerDataHeader
typedef struct _DBGKD_DEBUG_DATA_HEADER
{
     LIST_ENTRY64            List;
     ULONG                   OwnerTag;
     ULONG                   Size;

} DBGKD_DEBUG_DATA_HEADER, *PDBGKD_DEBUG_DATA_HEADER;


// Описание структуры ядра nt!KdDebuggerDataBlock
typedef struct _KDDEBUGGER_DATA
```

```
{
     DBGKD_DEBUG_DATA_HEADER  Header;

// Список переменных, общих для всех ОС платформы Windows NT

     ULONG64                  KernelBase;
     ULONG64                  BreakpointWithStatus;
     ULONG64                  SavedContext;
     USHORT                   ThCallbackStack;
     USHORT                   NextCallback;
     USHORT                   FramePointer;
     USHORT                   PaeEnabled:1;
     ULONG64                  KiCallUserMode;
     ULONG64                  KeUserCallbackDispatcher;
     ULONG64                  PsLoadedModuleList;
     ULONG64                  PsActiveProcessHead;
     ULONG64                  PspCidTable;
     ULONG64                  ExpSystemResourcesList;
     ULONG64                  ExpPagedPoolDescriptor;
     ULONG64                  ExpNumberOfPagedPools;
     ULONG64                  KeTimeIncrement;
     ULONG64                  KeBugCheckCallbackListHead;
     ULONG64                  KiBugcheckData;
     ULONG64                  IopErrorLogListHead;
     ULONG64                  ObpRootDirectoryObject;
     ULONG64                  ObpTypeObjectType;
     ULONG64                  MmSystemCacheStart;
     ULONG64                  MmSystemCacheEnd;
     ULONG64                  MmSystemCacheWs;
     ULONG64                  MmPfnDatabase;
     ULONG64                  MmSystemPtesStart;
     ULONG64                  MmSystemPtesEnd;
     ULONG64                  MmSubsectionBase;
     ULONG64                  MmNumberOfPagingFiles;
     ULONG64                  MmLowestPhysicalPage;
     ULONG64                  MmHighestPhysicalPage;
     ULONG64                  MmNumberOfPhysicalPages;
     ULONG64                  MmMaximumNonPagedPoolInBytes;
     ULONG64                  MmNonPagedSystemStart;
     ULONG64                  MmNonPagedPoolStart;
     ULONG64                  MmNonPagedPoolEnd;
     ULONG64                  MmPagedPoolStart;
```

```
ULONG64                 MmPagedPoolEnd;
ULONG64                 MmPagedPoolInformation;
ULONG64                 MmPageSize;
ULONG64                 MmSizeOfPagedPoolInBytes;
ULONG64                 MmTotalCommitLimit;
ULONG64                 MmTotalCommittedPages;
ULONG64                 MmSharedCommit;
ULONG64                 MmDriverCommit;
ULONG64                 MmProcessCommit;
ULONG64                 MmPagedPoolCommit;
ULONG64                 MmExtendedCommit;
ULONG64                 MmZeroedPageListHead;
ULONG64                 MmFreePageListHead;
ULONG64                 MmStandbyPageListHead;
ULONG64                 MmModifiedPageListHead;
ULONG64                 MmModifiedNoWritePageListHead;
ULONG64                 MmAvailablePages;
ULONG64                 MmResidentAvailablePages;
ULONG64                 PoolTrackTable;
ULONG64                 NonPagedPoolDescriptor;
ULONG64                 MmHighestUserAddress;
ULONG64                 MmSystemRangeStart;
ULONG64                 MmUserProbeAddress;
ULONG64                 KdPrintCircularBuffer;
ULONG64                 KdPrintCircularBufferEnd;
ULONG64                 KdPrintWritePointer;
ULONG64                 KdPrintRolloverCount;
ULONG64                 MmLoadedUserImageList;
ULONG64                 NtBuildLab;
ULONG64                 KiNormalSystemCall;
ULONG64                 KiProcessorBlock;
ULONG64                 MmUnloadedDrivers;
ULONG64                 MmLastUnloadedDriver;
ULONG64                 MmTriageActionTaken;
ULONG64                 MmSpecialPoolTag;
ULONG64                 KernelVerifier;
ULONG64                 MmVerifierData;
ULONG64                 MmAllocatedNonPagedPool;
ULONG64                 MmPeakCommitment;
ULONG64                 MmTotalCommitLimitMaximum;
ULONG64                 CmNtCSDVersion;
ULONG64                 MmPhysicalMemoryBlock;
ULONG64                 MmSessionBase;
```

```
    ULONG64           MmSessionSize;
    ULONG64           MmSystemParentTablePage;


// Список переменных, добавленных в ОС Windows Server 2003
(NT 5.2)

    ULONG64           MmVirtualTranslationBase;
    USHORT            OffsetKThreadNextProcessor;
    USHORT            OffsetKThreadTeb;
    USHORT            OffsetKThreadKernelStack;
    USHORT            OffsetKThreadInitialStack;
    USHORT            OffsetKThreadApcProcess;
    USHORT            OffsetKThreadState;
    USHORT            OffsetKThreadBStore;
    USHORT            OffsetKThreadBStoreLimit;
    USHORT            SizeEProcess;
    USHORT            OffsetEprocessPeb;
    USHORT            OffsetEprocessParentCID;
    USHORT            OffsetEprocessDirectoryTableBase;
    USHORT            SizePrcb;
    USHORT            OffsetPrcbDpcRoutine;
    USHORT            OffsetPrcbCurrentThread;
    USHORT            OffsetPrcbMhz;
    USHORT            OffsetPrcbCpuType;
    USHORT            OffsetPrcbVendorString;
    USHORT            OffsetPrcbProcStateContext;
    USHORT            OffsetPrcbNumber;
    USHORT            SizeEThread;
    ULONG64           KdPrintCircularBufferPtr;
    ULONG64           KdPrintBufferSize;
    ULONG64           KeLoaderBlock;
    USHORT            SizePcr;
    USHORT            OffsetPcrSelfPcr;
    USHORT            OffsetPcrCurrentPrcb;
    USHORT            OffsetPcrContainedPrcb;
    USHORT            OffsetPcrInitialBStore;
    USHORT            OffsetPcrBStoreLimit;
    USHORT            OffsetPcrInitialStack;
    USHORT            OffsetPcrStackLimit;
    USHORT            OffsetPrcbPcrPage;
    USHORT            OffsetPrcbProcStateSpecialReg;
    USHORT            GdtR0Code;
```

```
    USHORT              GdtR0Data;
    USHORT              GdtR0Pcr;
    USHORT              GdtR3Code;
    USHORT              GdtR3Data;
    USHORT              GdtR3Teb;
    USHORT              GdtLdt;
    USHORT              GdtTss;
    USHORT              Gdt64R3CmCode;
    USHORT              Gdt64R3CmTeb;
    ULONG64             IopNumTriageDumpDataBlocks;
    ULONG64             IopTriageDumpDataBlocks;


// Список переменных, добавленных в ОС Windows Vista (NT 6.0)

    ULONG64             VfCrashDataBlock;
    ULONG64             MmBadPagesDetected;
    ULONG64             MmZeroedPageSingleBitErrorsDetect
ed;


// Список переменных, добавленных в ОС Windows 7 (NT 6.1)

    ULONG64             EtwpDebuggerData;
    USHORT              OffsetPrcbContext;

} KDDEBUGGER_DATA, *PKDDEBUGGER_DATA;
```

Листинг 1. Описания необходимых внутренних структур ядра

Необходимо отметить тот факт, что поля, содержащие адреса глобальных переменных, выровнены на 64 бита. Это говорит о том, что данные структуры ядра можно использовать как в 64-разрядных (x64) версиях ОС, так и в 32-разрядных (x86) версиях ОС (просто отбрасывая 32 верхних разряда адреса).

Описав необходимые структуры ядра, можно приступать к реализации алгоритма поиска. Напишем функцию, получающую адреса необходимых глобальных переменных ядра, обращаясь к полям описанных структур ядра. Далее приведен листинг 2 с примером реализации подобной функции.

```
VOID    KeGetGlobalKernelVariables()
{
   KPCR                          *kpcr;
   DBGKD_GET_VERSION             *kdvb;
   DBGKD_DEBUG_DATA_HEADER       *kddbgh;
   KDDEBUGGER_DATA               *kddbgdb;

   // Получаем адрес структуры KPCR текущего процессора
   __asm
   {
      mov      eax, dword ptr fs:0x1c;
   mov  kpcr, eax;
   };

   NT_KPCR = kpcr;

   // Получаем адрес структуры DBGKD_GET_VERSION
   kdvb = kpcr->KdVersionBlock;

if (kdvb)
   {
      NT_KD_VERSION_BLOCK = kdvb;

      // Получаем адрес структуры DBGKD_DEBUG_DATA_HEADER
      kddbgh = kdvb->DebuggerDataList;

      // Получаем адрес структуры KD_DEBUGGER_DATA
      kddbgdb = kddbgh->List.Flink;

      if (kddbgdb)
      {

         // Получаем адреса необходимых переменных ядра
         NT_KD_DEBUGGER_DATA_BLOCK = kddbgdb;
         NT_KERNEL_BASE = kddbgdb->KernelBase;
         NT_PSP_CID_TABLE = kddbgdb->PspCidTable;
   NT_PS_ACTIVE_PROCESS_HEAD = kddbgdb->PsActiveProcessHead;
      NT_PS_LOADED_MODULE_LIST = kddbgdb->PsLoadedModuleList;
         // ...Список других необходимых переменных
      };
   };
};
```

Листинг 2. Функция поиска адресов глобальных переменных ядра

После выполнения такой функции в объявленных переменных вида **NT_XXX** будут содержаться адреса необходимых глобальных переменных ядра ОС.

Однако у приведенного выше варианта функции есть один небольшой недостаток: на многопроцессорных системах результат выполнения такой функции может быть некорректным, т. е. полученные адреса могут оказаться нулевыми, что может привести к краху системы при попытке обращения к ним. Это связано с тем, что действительный адрес структуры **nt!KdVersionBlock** находится только в той структуре **KPCR**, что описывает первый процессор в системе (под номером 0). Для решения данной проблемы требуется выполнение кода функции поиска именно на первом процессоре в системе. Это можно осуществить несколькими способами, например, используя механизм отложенного вызова процедур (Deferred Procedure Call – DPC). Данный механизм позволяет запускать выполнение кода на конкретном процессоре с конкретными параметрами, добавляя указанную процедуру в очередь на выполнение указанному процессору.

В данной работе будет использован чуть более простой способ, а точнее установка битовой маски **Affinity** для имеющихся в системе процессоров. Эта битовая маска представляет собой последовательность бит, размерностью совпадающую с разрядностью ОС. Каждый бит определяет соответствующий номер процессора. С помощью данной битовой маски устанавливается набор процессоров, которые будут задействованы для выполнения кода заданного потока. Воспользуемся этим способом для того, чтобы код функции поиска адресов глобальных переменных ядра выполнялся только на первом процессоре, структура **KPCR** которого содержит действительный адрес в поле **KdVersionBlock**. На листинге 3 представлен соответствующий код.

```
// Установка битовой маски
KeSetSystemAffinityThread(1);

// Вызов функции поиска глобальных переменных ядра
KeGetGlobalKernelVariables();

// Восстановление оригинальной битовой маски
KeRevertToUserAffinityThread();
```

Листинг 3. Гарантированное выполнение функции поиска на первом процессоре

Разработанный модуль ядра для динамического поиска адресов глобальных переменных ядра был протестирован на следующих 32-разрядных версиях ОС семейства Windows NT:

- Microsoft Windows XP (NT 5.1);
- Microsoft Windows Server 2003 (NT 5.2);
- Microsoft Windows Vista (NT 6.0);
- Microsoft Windows 7 (NT 6.1).

Для каждой из представленных ОС были проведены серии тестов, среди которых:

- запуск в ОС с различным набором установленных пакетов обновлений;
- запуск в ОС, установленных на машинах с различным аппаратным обеспечением (в том числе на одноядерных и многоядерных процессорах);
- запуск в ОС, установленных на виртуальных машинах (в том числе на одноядерных и многоядерных процессорах).

Разработанный модуль ядра может быть использован для динамического поиска адресов глобальных переменных ядра современных ОС семейства Windows NT как на однопроцессорных (Uni-Processor – UP), так и на многопроцессорных (Multi-Processor – MP) системах 32-разрядной архитектуры, например на серверах и рабочих станциях распределенных информационных систем. Представленный исходный код требует минимальной корректировки для возможности использования его в системах 64-разрядной архитектуры. Данный модуль может быть в дальнейшем использован различными программными средствами защиты, например антивирусными средствами или системами класса HIPS, для обнаружения и активного противодействия вредоносным программам.

Заключение

В ходе данной работы были решены следующие поставленные задачи:

- представлено теоретическое обоснование возможности создания нового способа нахождения глобальных переменных ядра;
- представлено экспериментальное обоснование возможности практического использования разработанного способа.

На основании полученных результатов были сделаны следующие выводы:

- все современные ОС семейства Windows NT содержат внутренние структуры, предназначенные для использования отладчиком ядра;
- эти структуры предоставляют актуальную и корректную информацию о текущем состоянии системы в любой момент времени;
- разработанный способ динамического поиска глобальных переменных ядра является наиболее эффективным среди существующих.

Разработанный в данной работе способ динамического поиска глобальных переменных ядра позволяет упростить и ускорить разработку средств защиты информации, а также решить ряд проблем, возникающих при разработке современных программных средств защиты для распределенных информационных систем, построенных на базе ОС семейства Windows NT.

Примечания

1 См.: *Грушо А.А., Тимонина Е.Е.* Распределенные атаки на распределенные системы // Информационный бюллетень JET INFO. М., 2006.

2 См.: *Хоглунд Г., Батлер Дж.* Руткиты: внедрение в ядро Windows. СПб.: Питер, 2007.

3 См.: *Barbosa E.* Finding some non-exported kernel variables in Windows XP [Electronic data]. [2004]. URL: http://www.reverse-engineering.info/System Information/GetVarXP.pdf

4 См.: *Руссинович М., Соломон Д.* Внутреннее устройство Microsoft Windows: Windows Server 2003, Windows XP и Windows 2000: Пер. с англ. 4-е изд. СПб.: Питер, 2008.

5 См.: *Russinovich M., Solomon D., Ionescu A.* Windows Internals: Covering Windows Server 2008 and Windows Vista, 5-th edition. Microsoft Press, 2009.

И.Г. Казовский

ПЛАТФОРМА ДЛЯ ПОСТРОЕНИЯ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ, ОСНОВАННЫХ НА ПРАВИЛАХ

Цель статьи – описание программы, представляющей собой универсальную платформу для разработки, отладки и сравнения алгоритмов машинного обучения, основанных на правилах. В статье рассматриваются общие вопросы организации платформы, ограничения, накладываемые на выполнимые в рамках данной платформы алгоритмы. Автором также приводится пример обучения системы при помощи ДСМ-метода на двух разных типах объектов.

Ключевые слова: платформа, алгоритмы машинного обучения, основанные на правилах, системонезависимость, инвариантность относительно типа данных, инвариантность относительно алгоритма.

Введение

Написанная программа представляет собой интегрированную платформу для построения систем анализа данных, основанных на правилах[1]. Платформа предоставляет объектную модель для реализации как систем анализа данных, так и структуры самих данных. Это позволяет проводить эксперименты не только с различными системами, но и с различными представлениями данных при работе с одной системой.

Нельзя сказать, что раньше подобных программ не создавалось. Но ни одна из них не претендует на такую степень универсальности. Существующие аналоги почти всегда основаны на готовом наборе алгоритмов. В основном этот набор исчерпывающий и добавить новые алгоритмы нельзя, хотя бывают исключения (например, ASDIEL, но его рассмотрим особо). Также большинство аналогичных систем накладывают какие-либо ограничения как на структуру данных, так и на системы анализа данных.

Также следует отметить, что практически все подобные программы либо являются непереносимыми, либо их надо собирать для каждой конкретной платформы (кроме написанных на языке Java). Представленная в данной работе платформа написана на языке C# и разработана под mono, что делает ее почти столь же переносимой, как и программы, написанные на языке Java. С другой стороны, среда исполнения mono исполняет код IL, что позволяет писать модули к платформе на любом языке высокого уровня, компилируемом в IL.

Предшественники

1. ASDIEL

Система ASDIEL ориентирована на подбор композиции алгоритмов для решения какой-либо конкретной задачи классификации или распознавания[2]. В системе реализовано множество алгоритмов, и она нацелена на работу в основном с ними.

Для запуска пользовательского модуля в данной системе необходимо написать динамическую библиотеку на языке C или C++. Этот подход порождает множество неудобств при необходимости отладки пользовательского модуля.

Решение задачи в системе ASDIEL начинается с выделения нескольких упорядоченных множеств, называемых *наборами*, которые в дальнейшем будут играть роль размерностей в массивах данных. Если предполагается работать только с признаковыми описаниями объектов, то достаточно определить лишь два набора: объектов и признаков. В общем случае могут понадобиться также:

- набор пар объектов;
- набор моментов времени (в динамических задачах);
- набор функций над парами объектов (расстояния, отношения и т. д.);
- набор свойств признаков (информативности, средние значения и т. д.);
- набор функций над парами признаков (близости, корреляции и т. д.);
- другие наборы, необходимость которых обусловлена особенностями конкретной задачи.

Затем декартовым произведениям некоторых из введенных наборов сопоставляются массивы, предназначенные для хранения и представления данных[2].

Спорным моментом системы ASDIEL можно считать то, что она обладает собственным интерпретируемым языком для написания

сценариев композиции алгоритмов. Здесь можно найти как положительные стороны – для изменения параметров композиции не требуется перекомпилировать весь модуль, так и отрицательные – как и любой другой язык программирования, язык ASDIEL имеет свои ограничения, и, когда будет достигнут его предел, придется вносить изменения в модуль.

2. QuDA

Система QuDA осуществляет интеллектуальный анализ данных по встроенным алгоритмам анализа формальных понятий, ДСМ-метода и др.[3,4] Каждый алгоритм имеет свои настройки, которые пользователь может изменить, открыв соответствующую форму. Обилие встроенных алгоритмов порождает обилие форм настроек этих алгоритмов, что либо увеличивает размер самой программы, либо усложняет код.

Еще один серьезный недостаток системы QuDA – при таком сложном интерфейсе руководство пользователя слишком лаконично и не включает описания всех функций, параметров настройки и особенностей программы.

Система написана на языке Java, что можно отнести к достоинствам. Язык Java является действительно кроссплатформенным, однако сама система не предусматривает дополнений и расширений. Пользователь может добавлять новую функциональность только встраивая ее в проект на этапе сборки.

3. Weka[5]

Система Weka, как и QuDA, написана на языке Java, однако в отличие от нее, имеет подробное руководство пользователя.

К плюсам системы Weka можно также отнести модульность – система предоставляет пользователю возможность написать свой алгоритм анализа данных и встроить его в систему во время исполнения (run-time).

Однако для отладки этого алгоритма придется перекомпилировать пользовательский модуль и перезапустить всю систему, что довольно утомительно, особенно в случае обнаружения случайных мелких ошибок. Наличие в системе поддержки интерпретируемого языка позволило бы избавиться от этой проблемы.

4. Прочие решатели

Существует большое количество реализаций отдельных систем анализа данных, но они нацелены на решение одной конкретной задачи или класса задач, поэтому мы их не будем здесь рассматривать.

Описание системы анализа данных

Поскольку каждая система анализа данных работает с множеством данных, то среда содержит подсистему управления множеством объектов, и, чтобы реализовать новую систему анализа данных, необходимо задать только алгоритм работы с этим множеством. Для описания этого алгоритма можно попробовать обойтись настройками подсистемы управления множеством объектов только через графический интерфейс. Но тогда либо интерфейс будет чрезмерно сложен, либо класс алгоритмов, реализуемых в рамках данной среды, будет слишком узким. Поэтому возможны два варианта – использовать модули расширения среды или встроить в систему интерпретируемый язык подсистемы управления данными. Сама среда написана на языке C#, что позволяет писать модули на языках высокого уровня (на любом языке, компилируемом в IL).

Средствами API невозможно добиться такой же гибкости, как скриптовым языком. Чтобы сделать среду максимально универсальной, было решено использовать уже существующую систему управления данными. Наиболее естественный путь – встроить в среду какую-либо СУБД. Это возможно, но сильно увеличивает размер программы и ее требования к аппаратной части. Кроме того, языки управления СУБД – процедурные (в большей или меньшей степени), а это усложняет описание правил в данном языке.

Можно написать свой язык сценариев и, соответственно, свой интерпретатор этого языка. Однако это нерационально по двум причинам:

- разработка адекватного языка (и его интерпретатора) – задача далеко не тривиальная и довольно продолжительная по времени;
- новый язык следует писать, учитывая опыт предшественников, т. е. он будет похож на существующие сценарные языки. С другой стороны, в языке должны будут присутствовать конструкции, специфические для данной среды.

Гораздо удобнее со всех точек зрения – встроить в среду язык логического программирования (такой, как Prolog или CLIPS).

Интерпретатор языка Prolog осуществляет резолютивный вывод в логике дизъюнктов Хорна методом обратного логического вывода (backward chaining), т. е. от резольвенты к начальным посылкам[6]. Prolog, как и многие системы с обратным выводом, не сохраняет промежуточных результатов. Это существенно экономит память, но очень плачевно сказывается на времени выполнения программы. Такие особенности языка Prolog накладывают ограничения на область применения использующей его среды.

В обсуждаемую в данной работе среду встроен скриптовый язык, а именно CLIPS[7]. Он позволяет задавать разные системы вывода (на классической и неклассической логике), что только добавляет гибкость среде. В отличие от языка Prolog и его потомков, интерпретатор языка CLIPS осуществляет вывод методом прямых рассуждений (forward chainig), сохраняя промежуточные результаты, что, конечно, занимает память, но увеличивает производительность (иногда довольно существенно).

При описании правил вывода в логике предикатов часто определяются новые операции над объектами. Интерпретатор CLIPS допускает определение новых, пользовательских, функций (User-Function[8]). Кроме того, язык CLIPS имеет объектно-ориентированное расширение – COOL. В предлагаемой среде эти возможности сохранены. Таким образом, описание правил на языке CLIPS может содержать как описание новых операций (на языке CLIPS, например, с помощью конструкции *deffunction*, или в .NET-модуле через API среды), так и описание типов данных (также либо на расширении языка CLIPS – на языке COOL, либо во внешнем модуле на .NET-языке).

Далее, под языком *CLIPS* будем понимать как собственно CLIPS, так и его расширение COOL. Реализацию .NET-модуля, расширяющего язык CLIPS, будем называть *CLIPS-модуль*, а описание правил на языке CLIPS и модуля (возможно пустого) – *стратегией*.

Модельная задача для такой среды – ДСМ-метод порождения гипотез. Метод применим для широкого класса задач и использует правдоподобные рассуждения при порождении гипотез. Правила рассуждений описываются на языке логики предикатов и легко переписываются на язык CLIPS. Для проверки работы метода можно использовать объекты с примитивной структурой и простым поведением.

Пример системы анализа данных

В качестве примера рассмотрим ДСМ-метод порождения гипотез (как уже отмечалось выше, это модельная задача для среды), предложенный В.К. Финном[9]. Подробное описание ДСМ-метода также можно найти в работах[10, 11].

В рамках данной работы для краткости опишем только алгоритм упрощенного ДСМ-метода без запрета на контрпример. Для описания алгоритма сделаем несколько допущений. Пусть

- **O** – коллекция объектов обучения;
- **P** – коллекция целевых свойств;

- **S** – коллекция возможных причин наличия / отсутствия свойств объектов (далее для краткости будем считать, что возможные причины – элементы структуры объектов);
- у класса «объект обучения» определен метод Similarity, возвращающий результат операции сходства двух объектов – возможную причину наличия / отсутствия свойств(а);
- класс «возможная причина» определяет метод IsIn, который возвращает true, если данная причина содержится в объекте, переданном в качестве аргумента, и false в противном случае;
- определено перечисление InternalTruthValues (типы внутренних истиностных значений) как:
  True = +1 (фактически истинно);
  False = –1 (фактически ложно);
  Contradictory = 0 (фактически противоречиво);
  Uncertain = ? (не определено);
- у классов «объект обучения» и «возможная причина» определен оператор индексирования ([,]), который в качестве параметра принимает целевое свойство и возвращает или устанавливает значение типа InternalTruthValues.

Теперь определим алгоритмы вывода и запишем их на псевдокоде следующим образом.

1. Алгоритм порождения гипотез о возможных причинах (индукция)

```
foreach o1 in O {
   foreach o2 in O where o1 != o2 {
   foreach p in P {
       if(o1[p] == o2[p] && o1[p] != Uncertain) {
       s = o1.Similarity(o2);
       foreach o in O where o != o1 && o != o2 {
       if(o[p]!= Uncertain &&
            o[p] != o1[p] && s.IsIn(o)) {
                    s[p]= Contradictory;
                    gotoNextStep;
          }}
          s[p]= o1[p];
   NextStep :
          S.Add(s);
          }}}}
```

2. Алгоритм порождения гипотез о наличии или отсутствии целевых свойств (аналогия)

```
foreach s in S {
   foreach p in P where s[p] != Uncertain {
   foreach o in O where o[p] == Uncertain {
   if(s.IsIn(o)) {
       foreach s1 in S where s != s1 && s[p] != s1[p] {
            if(s1 .IsIn(o)) gotoContr;
            }}
            o[p] = s[p];
            continue;
Contr :
            o[p] = Contradictory;
       }}}
```

Каждый алгоритм работает до тех пор, пока он вносит в систему изменения. Потом запускается другой алгоритм. Когда ни один из них не может внести изменений, считается, что система закончила свою работу.

1. Стратегия

Опишем на языке CLIPS правила обучения. Для начала определим шаблон объекта и причины:

```
(deftemplate Object
(slot id)
(slot data-type)
(multislot properties)
)
```

Слот id будет содержать объект (точнее ссылку на .Net-части описания объекта), data-type – SYMBOL, указывающий тип факта (объект или причина), мультислот properties – множество свойств объекта.

Правила, записанные на языке CLIPS, будут выглядеть следующим образом:

```
(defrule reason-search
   (Fact (id ?x1) (data-type object))
   (Fact (id ?x2) (data-type object))
   (test (neq ?x1 ?x2))
=>
   (bind ?x (similarity ?x1 ?x2))
   (if (neq ?x FALSE) then
   (bind $?p (get-properties ?x))
```

```
      (assert
      (Fact (id ?x)(data-type peculiarity)(properties
      $?p)
      )
   )
)
(defrule refill-knowledge-base
   ?f <- (Fact (id ?x) (data-type object)(properties
   $?p))
   (Fact (id ?y) (data-type peculiarity))
   (test (is-in ?y ?x))
=>
   (bind $?p (change-properties ?x ?y))
   (modify ?f (properties $?p))
)
```

Функции `similarity`, `get-properties`, `is-in`, `change-property`, очевидно, являются функциями пользователя. Реализацию их прототипов переложим на C#-модуль.

Еще один момент, который был сознательно опущен выше, но который необходимо упомянуть – каждый объект, удовлетворяющий первому шаблону из левой части правила reason-search, удовлетворяет и второму шаблону, и наоборот. То есть для двух разных объектов o_1 и o_2 система CLIPS дважды применит правило reason-search – один раз для пары (o_1, o_2), а второй раз для пары (o_2, o_1). Чтобы этого не происходило, введем еще один тип факта и изменим правило reason-search следующим образом:

```
(deftemplate Analyzed
   (slot first)
   (slot second)
)
(defrule reason-search
   (Fact (id ?x1) (data-type object))
   (Fact (id ?x2) (data-type object))
   (not (exists (Analyzed (first ?x2)(second ?x1))))
   (test (neq ?x1 ?x2))
=>
   (assert (Analyzed (first ?x1)(second ?x2)))
   (bind ?x (similarity ?x1 ?x2))
   (if (neq ?x FALSE) then
   (bind $?p (get-properties ?x))
     (assert
```

```
      (Fact (id ?x)(data-type peculiarity)(properties
      $?p))
      )
   )
)
```

2. .Net ДСМ-модуль

Прежде всего необходимо прототипировать внешние для интерпретатора CLIPS функции. Чтобы система анализа данных корректно встроила прототипы функций, CLIPS-модуль должен содержать реализацию некоторых интерфейсов и наследников классов.

Класс, расширяющий базовый набор функций языка CLIPS, должен наследоваться от класса `BaseClasses.Education.Engine` и реализовывать два абстрактных метода: `RegistrateFunctions` и `UnregistrateFunctions`. Первый метод вызовется автоматически в конструкторе базового класса, поэтому, в частности, в конструктор передается экземпляр класса `BaseClasses.CLIPS.CLIPSProvider` – обертка (wrapper) вокруг .NET окружения CLIPS (`CLIPSNet.Environment`). Второй метод вызывается тоже автоматически, при вызове метода `BaseClasses.Education.Engine.Dispose()`.

Чтобы зарегистрировать обработчик новой функции пользователя, необходимо вызвать метод RegistrateFunction экземпляра класса `CLIPSNet.Environment`. Первый параметр – имя функции, второй – делегат (`System.Delegate`), реализующий саму функцию. Указанный делегат должен принимать параметры только типов, унаследованных от типа `CLIPSNet.CLIPSDataTypes.DataType`, и возвращать значение только типа, реализующего интерфейс `CLIPSNet.CLIPSDataTypes.IDataType`.

2.1. Similarity

Исходя из вышеописанных правил функция Similarity должна возвращать символ FALSE, если сходство объектов пустое, или идентификатор созданного объекта – причины. В варианте библиотеки CLIPS, написанной на языке C, есть возможность передать в окружение CLIPS указатели на внешние объекты. В библиотеке CLIPSNet подобная функциональность инкапсулирована в классе `CLIPSDataTypes.ExternalObject`. Описание метода Similarity на C# будет выглядеть так:

```
public IDataType Similarity(ExternalObject eo1,
                                  ExternalObject eo2)
{
```

```
JSM.Object o1 = (JSM.Object)eo1.Value;
JSM.Object o2 = (JSM.Object)eo2.Value;
JSM.Peculiarity o = o1.Similarity (o2);
if(o == null || peculiarities.Contains(o))
return new Symbol(«FALSE»);
peculiarities.Add(o); //peculiarities – список причин
return new ExternalObject(o);
}
```

А регистрация этой пользовательской функции в окружении CLIPS так:

```
protected override void RegistrateFunctions()
{
   CLIPSNet.Environment   env   =   CLIPSProvider.
   Environment;
   env.RegistrateFunction(«similarity»,
                    new LikeDelegate(Similarity));
}
```

Из приведенного выше описания метода Similarity следует требование к классу JSM.Object – он должен содержать определение операции сходства. Эта операция должна выдавать причину свойств объектов, если свойства определены и равны у обоих объектов, или null в противном случае.

2.2. GetProperties

Исходя из описания стратегии эта функция должна возвращать список свойств объекта. Поскольку свойств может быть несколько, то делегат, реализующий эту функцию, должен возвращать объект типа `CLIPSNet.CLIPSDataTypes.Multifield` – коллекцию объектов, реализующих интерфейс `CLIPSNet.CLIPSDataTypes.IDataType`.

```
public Multifield GetProperties(ExternalObject eo)
{
   Object o = (Object)eo.Value;
   PropertyCollection pc = o.Properties;
   Multifield mf = new Multifield();
   foreach(IProperty p in pc)
   {
   mf.Add(new Symbol(p.Name));
       mf.Add(new CLIPSNet.CLIPSDataTypes.
```

```
                              String(p.ValueToString()));
   }
   return mf;
}
```

Нетрудно увидеть, что объект записывается в Multifield в виде пар «имя – значение».

2.3. IsIn

IsIn – метод, позволяющий определить, содержится ли возможная причина в объекте. А поскольку на данном этапе разработки ДСМ-системы нам ничего не известно про структуру объектов, то оставим определение этого метода реализациям прикладных систем.

```
public  CLIPSDataTypes.Boolean  IsIn(ExternalObject
eo1,
                                     ExternalObject eo2)
{
   Peculiarity p = (Peculiarity)eo1.Value;
   Object o = (Object)eo2.Value;
   return   new   CLIPSNet.CLIPSDataTypes.Boolean(p.
   IsIn(o));
}
```

2.4. ChangeProperty

Эта функция изменяет свойства объекта и возвращает все свойства объекта в виде списка (`multifield` в CLIPS или `CLIPSNet.DataTypes.Multifield` – в C#).

```
public Multifield ChangeProperty(ExternalObject eo1,
                                     ExternalObject eo2)
{
   Object o = (Object)eo1.Value;
   Peculiarity p = (Peculiarity)eo2.Value;
   foreach(IProperty ip in p.Properties)
   {
   ((Property)o.Properties[ip.Name]).
                        ChangeTo(ip.ValueToString());
       PropertyCollection pc = o.Properties;
       Multifield mf = new Multifield();
       foreach(IProperty p in pc)
       {
       mf.Add(new Symbo1(p.Name));
```

```
        mf.Add(new CLIPSNet.CLIPSDataTypes.String(
                                    p.ValueToString()));
        }
        return mf;
    }
}
```

2.5. JSM.Object и JSM.Peculiarity

Теперь, исходя из требований, предъявленных выше к классам JSM.Object и JSM.Peculiarity, можно описать их структуру примерно так (ограничимся словесным описанием второстепенных методов и классов или вовсе опустим их):

```
public interface ISimilar
{
   ISimilar Similarity(ISimilar il);
}
public abstract class Object : ISimilar
{
   protected PropertyCollection properties;
   public PropertyCollection Properties
   {
   get { return properties; }
   }
   public Object()
   {
   properties = new PropertyCollection();
   }
   public abstract string ValueToString();
   public ISimilar Like(ISimilar c)
   {
   return Similarity((Object)c);
   }
   public abstract Peculiarity Similarity(Object o);
}

public abstract class Peculiarity : Object
{
   public abstract bool IsIn(Object o);
}
```

Класс `Property` – описывает целевое свойство, а `Property Collection` – множество целевых свойств. Поскольку значения це-

левых свойств могут быть разных типов, заранее неизвестных, то сам класс `Property` – `Generic`-класс, а класс `PropertyCollection` содержит элементы типа интерфейса `IProperty` (который, в частности, реализуется классом `Property`).

Внутренние истиностные значения ДСМ-метода определим в виде перечисления:

```
public enum InternalValues
{
   False = (int)'-',
   True = (int)'+',
   Contradictory = (int)'0',
   Uncertain = (int)'?'
}
```

Тестовые задачи

1. Определение рода слова

Построив модель, теперь перейдем к описанию данных. Пусть дано множество слов. Слова могут быть только женского или неженского рода. Пусть в этом множестве есть несколько слов, помеченных как слова женского рода, и несколько слов, помеченных как слова неженского рода. Определим операцию сходства двух слов, результат выполнения которой равен последней букве этих слов, если они совпадают, в противном случае FALSE. Задача состоит в том, чтобы найти все признаки рода и по ним определить род всех непомеченных слов (проект SexDerminer).

Пусть дан список слов, для которых известен род.

Слово	Женский род	Слово	Женский род
Palka	+	Oko	–
Caplja	+	Liko	–

А для слов luna, duga, reka, lozka, polka, stenka, ruchka, krishka, pulja, kopalka, veko, varevo, ruka, duga, steklo, dulja род не известен, и его надо определить.

Программа определяет, что слова женского рода оканчиваются на букву '**a**', а неженского рода – на '**o**', и, соответственно, по окончанию определяет род слов. Таким образом, в результате получается следующая таблица:

Слово	Женский род	Слово	Женский род
Palka	+	Oko	–
Caplja	+	Liko	–
Luna	+	Veko	–
Duga	+	Varevo	–
Reka	+	Ruka	+
Lozka	+	Duga	+
Polka	+	Cteklo	–
Stenka	+	Ruchka	+
Krishka	+	Pulja	+
Kopalka	+	Dulja	+

2. Определение тематики текста

Пусть дано множество текстов, каждый из которых состоит из пары ключевых слов. Про какие-то тексты известно, что они про оружие, про некоторые другие известно, что они не про оружие. Операцию сходства для двух текстов определим как общее ключевое слово, содержащееся в каждом тексте. Будем также считать, что если немаркированный текст содержит маркированное ключевое слово, то и весь текст становится маркированным. Задача – определить тематику каждого текста и найти и пометить все ключевые слова (проект WeaponTest).

Пусть известно про следующие тексты.

Текст	Про оружие	Текст	Про оружие
(bow, sword)	+	(mattok, hook)	–
(bow, pike)	+	(hook, plow)	–
(sword, pike)	+	(plow, mattok)	–

И пусть есть непомеченные тексты: (bow, crossbow), (sword, crossbow), (crossbow, smallsword), (sword, smallsword), (smallsword, musquet), (smallsword, pistol), (crossbow, musquet), (crossbow, pistol), (musquet, rifle), (pistol, rifle), (rifle, gun), (pistol, gun), (gun, machine-gun), (pistol, machine-gun), (pistol, revolver), (gun, revolver), (hook, scythe), (mattok, scythe), (mattok, plough), (plow, plough), (scythe, cropper), (plough, cropper), (plough, field-engine), (cropper, field-engine).

По завершении работы программа определяет тематику текстов:

Текст	Про оружие	Текст	Про оружие
(bow, sword)	+	(mattok, hook)	–
(bow, pike)	+	(hook, plow)	–
(sword, pike)	+	(plow, mattok)	–
(bow, crossbow)	+	(hook, scythe)	–
(sword, crossbow)	+	(mattok, scythe)	–
(crossbow, smallsword)	+	(mattok, plough)	–
(smallsword, musquet)	+	(scythe, cropper)	–
(smallsword, pistol)	+	(plough, cropper)	–
(crossbow, musquet)	+	(plough, field-engine)	–
(crossbow, pistol)	+	(cropper, field-engine)	–
(musquet, rifle)	+	(pistol, rifle)	+
(rifle, gun)	+	(pistol, gun)	+
(gun, machine-gun)	+	(pistol, machine-gun)	+
(pistol, revolver)	+	(gun, revolver)	+

Попутно с определением тематики текстов определяется и тематика ключевых слов.

Текст	Про оружие	Текст	Про оружие
plow	–	mattok	–
plough	–	hook	–
scythe	–	cropper	–
field-engine	–	pike	+
sword	+	bow	+
crossbow	+	smallsword	+
musquet	+	pistol	+
rifle	+	gun	+
machine-gun	+	revolver	+

Дальнейшее развитие

Следующий важный шаг в развитии программы – профилирование. Это необходимо для полноценного сравнения методов анализа данных. Причем замерять можно два разных аспекта

- *Производительность.* Понятно, что речь идет не только о тактах процессора, но и о сложности алгоритма, количестве выходов из среды CLIPS, количестве вызовов системных функций и т. д. Производительность должна измеряться по разным параметрам – не только системным. Будут реализованы API среды для профилирования пользовательских функций системы CLIPS и базовая подсистема профилирования выполнения правил анализа данных.
- *Семантическая полнота метода.* Это позволит сравнивать методы по тщательности анализа данных, по полноте множества порожденных гипотез, по их достаточности для объяснения всех фактов, по количеству лишних гипотез, порожденным методом. Конечно, в каждом методе по-своему определяются достаточность и полнота, поэтому будут описаны интерфейсы семантического профилирования, а конкретную реализацию придется делать в каждом конкретном случае.

Раз унифицируются способы задания структуры данных, то должна унифицироваться и процедура инициализации данных. То есть должна быть возможность для разных задач задать входные данные в одном формате. Соответственно, встает вопрос о подсистеме инициализации данных. Подсистема должна будет читать входной файл (поток, базу данных) и инициализировать базу фактов и, возможно, автоматически генерировать новые типы для описания и хранения введенных данных. После окончания анализа подсистема будет записывать результаты в выходной поток.

Ну и наконец, в дальнейшем графические средства платформы будут только улучшаться (в частности, в планах создание встроенного редактора для языка CLIPS и для .NET-модуля).

Поскольку, как уже говорилось выше, система написана на высокоуровневом языке программирования C#, то к среде можно дописывать модули на любом .NET-языке. Это позволяет создавать расширения среды практически любой степени сложности.

Примечания

[1] Исходные коды доступны по адресу: URL: http://sourceforge.net/projects/mlplatform (дата обращения: 20.12.2010).

[2] См.: Язык описания алгоритмических композиций ASDIEL. [Электронный ресурс] // Сайт «Распознавание, Классификация, Прогнозирование». [М., 2010]. URL: http://www.ccas.ru/frc/asdiel.html (дата обращения: 20.12.2010).

[3] См.: *Eklund Peter W.* Concept Lattices // Second International Conference on Formal Concept Analysis, ICFCA 2004, Sydney, Australia, February 23–26, 2004. Proceedings. 2004. Springer. 428 p.

[4] См.: *Grigoriev P.A., Yevtushenko S.A.* QuDA: Applying Formal Concept Analysis In a Data Mining Environment. Berlin: Springer, 2004.

[5] См.: *BouckaertR.R. , Frank E., Hall M., Kirkby R., Reutemann P., Seewald A., Scuse D.* WEKA Manual for Version 3-6-1. June 4. New Zealand, Hamilton: University of Waikato, 2009.

[6] См.: Пролог (язык программирования) [Электронный ресурс] // Сайт Википедия. [М., 2010]. URL: http://ru.wikipedia.org/wiki/Prolog (дата обращения: 20.12.2010).

[7] Библиотека CLIPSNet [Электронный ресурс] // Сайт Sourceforge. [USA, 2010]. URL: http://sourceforge.net/projects/clipsnet (дата обращения: 20.12.2010).

[8] См.: CLIPS Reference Manual Volume II Advanced Programming Guide. March 22, 2008.

[9] См.: ДСМ-метод. [Электронный ресурс] // Сайт Википедия. [М., 2010]. URL: http://ru.wikipedia.org/wiki/ДСМ-метод (дата обращения: 20.12.2010).

[10] *Финн В.К.* О машинно-ориентированной формализации правдоподобных рассуждений в стиле Ф. Бэкона – Д.С. Милля // Семиотика и информатика. 1983. Вып. 20. С. 35–101.

[11] *Аншаков О.М., Скворцов Д.П., Финн В.К.* О дедуктивной имитации некоторых вариантов ДСМ-метода автоматического порождения гипотез // Семиотика и информатика. 1993. Вып. 33. С. 164–233.

И.А. Хохряков

РЕАЛИЗАЦИЯ КОМПЛЕКСА ПРОГРАММНЫХ ИНСТРУМЕНТОВ ДЛЯ СОПРОВОЖДЕНИЯ ЭЛЕКТРОННЫХ ГРАММАТИЧЕСКИХ СЛОВАРЕЙ РУССКОЙ ЛЕКСИКИ

В статье описываются особенности программной реализации инструментальных приложений, облегчающих коррекцию и пополнение новыми словами специализированных электронных грамматических словарей. Словарь, рассматриваемый в данной статье, в первую очередь предназначен для поверхностно-синтаксического анализа русских текстов, однако может применяться и для решения других задач. Первоочередным критерием, положенным в основу данной реализации, является минимизация человеко-временных ресурсов, требуемых для разработки этих инструментов и их последующей модификации (в частности, их настройка на применение к сопровождению электронных словарей других форматов).

Ключевые слова: обработка естественно-языковых русских текстов, электронные словари для языков флективного типа, форматы словарей, поверхностно-синтаксический анализ текста, программная реализация, локальная и распределенная архитектуры программных приложений, реляционная БД, предметные XML языки, языки сценариев.

Как известно, синтаксический анализ является одним из необходимых средств решения задач, требующих понимания, в большей или меньшей степени, текста на естественном языке. К таким задачам относятся документальный (текстовый) поиск, извлечение знаний или хотя бы фактов из текста, перевод с одного языка на другой и т. п. Автоматизация решения подобных задач приводит к необходимости формального описания синтаксического анализа с целью его последующей программной реализации. Полнота такого формального описания существенно влияет на адекватность, корректность и точность конечного результата анализа текста, однако

«проблема синтаксического анализа неформализованных текстов в полном объеме еще не решена»[1].

В настоящее время в Институте лингвистики РГГУ проводится экспериментальная реализация системы правил поверхностно-синтаксического анализа (далее – ПСА) русского предложения, на протяжении многих лет разрабатываемой Т.Ю. Кобзаревой[2]. Ее подход отличается от подхода большинства других работающих в этой области исследователей тем, что синтаксический анализ рассматривается ею не во взаимодействии с семантическим анализом, а как самостоятельный этап анализа текста, предшествующий семантическому анализу и использующий минимальные семантические сведения. При таком подходе, вообще говоря, может возрастать число вариантов синтаксического разбора предложения, зато имеется возможность сосредоточить усилия именно на синтаксическом анализе, сужая круг исследуемых лингвистических ситуаций. Система ПСА открыта для пополнения новыми правилами и изменения уже имеющихся с целью ее совершенствования, что и является первоочередной задачей указанной реализации. Для обеспечения этого процесса была разработана специальная программная среда – экспериментальная система работы с лингвистическими алгоритмами (ЭСЛА)[3].

Наряду с правилами система ПСА эксплуатирует еще один важный информационный ресурс, а именно специализированный для решения этой задачи электронный грамматический словарь. В данной статье описывается инструментарий, дающий возможность эффективно пополнять этот словарь новыми словарными статьями, при необходимости исправлять имеющиеся статьи, осуществлять «учет и контроль» подобных изменений, а также кратко обсуждаются некоторые особенности программной реализации рассматриваемых инструментов.

Устройство словаря для поверхностно-синтаксического анализа

Разработка описываемого здесь электронного словаря была начата в 1970-е годы в «Информэлектро» в отделе, возглавляемом Д.Г. Лахути, группой[4] под руководством Г.А. Лесскиса[5]. Первая версия словаря использовалась как информационное обеспечение для морфологического анализатора «Скобки», который, в свою очередь, применялся в разрабатываемых группой системах обработки текстов на русском языке[6]; вторая версия[7] используется в настоящее время для разработки системы ПСА.

Содержательно словарь представляет систему грамматических категорий (часть речи, род, число, падеж, время и др.), управления (падежами, предлогами, инфинитивом, подчинительным союзом) и некоторых приписываемых словам семантических классов (неодушевленный предмет, одушевленный предмет, параметр и т. д.). Эти категории приписываются словам анализируемого предложения на стадии морфологического анализа (лемматизация), а затем уже используются как свойства слов в процессе применения правил ПСА.

Рассматриваемый словарь состоит из двух взаимосвязанных частей: словаря основ и словаря флексий (окончаний). (Под основой здесь понимается часть слова до окончания.) Грамматические категории и «системные» (вспомогательные) данные представлены в словарных статьях обоих словарей как свойства в виде пар «имя свойства; значение / список значений».

Словарь основ содержит индивидуальные описания лексем языка. Вследствие чередования лексема может содержать не одну, а несколько основ (например, *ветер* в им. и вин. падежах ед. числа, *ветр-* – в остальных формах: *ветра*, ...). Однако словарная статья этого словаря (отсюда его название) представляет только одну основу вместе с приписанными ей характеристиками – так называемый индивидуальный грамматический образ (далее – ИГО) основы. ИГО основы включает ссылку на ассоциированную с основой таблицу флексий (далее – ТФ), представленную в словаре окончаний. В такую таблицу входят все те окончания вместе с приписанными каждому их них свойствами (*ИГО окончаний*), которые могут быть дописаны к этой основе при словоизменении лексемы. Таким образом, объединение всех ТФ лексемы определяет ее словоизменительную парадигму.

В результате морфоанализа каждое анализируемое слово будет распознано как одна или несколько словоформ парадигмы одной или нескольких лексем. Например, *вершины* – это две формы (род. падежа ед. числа и им. падежа мн. числа) одного слова. А словоформа *стали* входит в разные парадигмы[8]: *выплавка стали* – существительное, *стали совсем никакими, стали ошибаться* – глагол. Словоформа с приписанными ей характеристиками, полученными комбинированием соответствующих ИГО основы и ИГО окончания, составляют *ИГО словоформы* (*ИГО слова в предложении*).

Такой двухчастный словарь, содержащий указанные выше грамматические признаки слов, будем далее называть ИГО-словарем, содержательный формат его статей – ИГО-форматом. ИГО-словарь может иметь разные форматы представления. В настоящее время используются:

– текстовый формат, в котором каждая из частей сохраняется в отдельном текстовом файле;
– соответствующий текстовому двоичный формат, который используется программой морфологического анализа;
– представление в виде реляционной БД;
– XML-формат представления ИГО-слова и ИГО-основы.

Задача пополнения ИГО-словаря и требования к инструментарию

В настоящее время ИГО-словарь включает около 30 тыс. слов. Этого достаточно для его использования в системе ЭСЛА с целью тестирования и отладки системы ПСА. При этом в случае необходимости в него время от времени добавляются новые слова из тестовых примеров. Однако даже для опробования системы на статистически значимом корпусе предложений этого количества недостаточно. Так, «Грамматический словарь русского языка» А.А. Зализняка[9], электронная версия которого используется во многих системах обработки русских текстов, содержит более 110 тыс. слов.

Словарная статья ИГО-словаря содержит некоторые данные, которые отсутствуют в словаре А.А. Зализняка (например, сведения об управлении, семантические классы), поэтому непосредственно применять последний в системе ПСА нельзя. Кроме того, всегда сохраняется необходимость оперативного добавления слов – новых слов в языке (особенно часто они появляются в Интернете), терминов при ориентации системы анализа на узкоспециализированную предметную область. Как и ПСА, словарь является открытой информационной системой.

При добавлении новой словарной статьи в ИГО-словарь возникают следующие сложности. Во-первых, нужно описать все актуальные для нового слова характеристики. Их перечень различен для разных групп слов и зависит от части речи. Но некоторые признаки для одних слов обязательны, а для других могут отсутствовать (имеют «нулевое» значение), причем иногда это определяется индивидуально для каждого слова. Например, для одних неотглагольных существительных управление указывается, а для других – нет (для отглагольных существительных эта характеристика обязательна). Во-вторых, необходимо правильно задавать значения таких свойств, выбирая их из области возможных значений для каждого признака.

Имеется справочная документация, содержащая облегчающие эту деятельность инструкции. Опыт показывает, что чем дольше лингвистически образованный специалист работает со словарем,

тем быстрее и качественнее он справляется с назначением почти всех, особенно предметно осмысленных, параметров. Однако конкретизация номера ТФ, соответствующей основе слова, является сложной рутинной задачей, трудоемкость которой слабо зависит от компетентности специалиста. Это связано с большим количеством ТФ (их насчитывается несколько сотен). Для решения этой задачи «вручную» имеются две возможности.

1. В словаре основ найти основу, соответствующую вновь заносимой и принадлежащую лексеме с такой же парадигмой, что и у добавляемого слова; взять номер ТФ из ИГО найденной основы. Данная процедура осложняется тем обстоятельством, что словарные статьи этого словаря упорядочены по основам в обратном (т. е. от «я» к «а») лексикографическом порядке. По этой причине у немалого количества слов основы «разбросаны по словарю», а не расположены подряд[10].

2. Воспользоваться так называемым указателем для соответствующей части речи. Этот указатель представляет собой вопросник, организованный в виде блок-схемы. Ее узлы «содержат» правила в виде условия-вопроса и нескольких вариантов ответов (наподобие тестов в обучающих системах). Выбор одного из ответов определяет переход к следующему узлу. Концевой узел блок-схемы, в котором мы закончим ее проход, содержит искомый номер ТФ. Общий объем этих указателей (только для существительных такая блок-схема занимает 29 печатных листов формата А4) во много раз превосходит объем остальной документации по ИГО-словарю[11].

Ясно, что задача заполнения ИГО-словаря «вручную» требует длительных усилий достаточно большого числа хорошо подготовленных специалистов.

Таким образом, становится актуальной автоматизация процесса пополнения ИГО-словаря с целью ускорения этого процесса и повышения его качества (в смысле корректности вводимых в словарь данных). Как нам представляется, комплекс программных приложений, разрабатываемых как инструментальные средства такой автоматизации, должен удовлетворять следующим требованиям.

1. Вся справочная информация, в том числе инструкции по пополнению словаря, должна стать электронным ресурсом, положенным в основу как алгоритмов, обеспечивающих функциональность программных инструментов, так и сценариев интерфейса пользователя, позволяющих гибко применять функциональные возможности. Необходимость обращения к документации в бумажном виде может возникать лишь на начальной стадии использования инструментов, во время обучения работе с ними.

2. Пользователю должны быть предоставлены возможности сочетать различные способы формирования словарной статьи – как универсальные (ввод значений параметров ИГО-основы «вручную из головы»; использование указателя для выбора номера ТФ), так и частные, существенно облегчающие задание каких-либо параметров не всегда, но хотя бы для достаточно часто встречающихся случаев (например, использующие информацию об аналогичных лексемах в словаре).

3. Указатели выбора номера ТФ представляют собой довольно сложную и вместе с тем открытую систему правил. Их коррекция и, при необходимости, пополнение новыми правилами «вручную» также являются трудоемкими задачами, поэтому требуется инструментальное средство для совершенствования указателей.

4. Дизайн интерфейса должен обеспечивать удобное и наглядное предъявление пользователю как редактируемых данных, так и сопутствующей информации, соответствующей выбранному пользователем методу построения новой словарной статьи. Аналогичные требования предъявляются и к интерфейсу инструмента для коррекции указателей.

5. Необходимы средства учета изменений, внесенных в словарь различными пользователями в разное время (*журнал изменений*), и контроля правильности этих изменений.

6. «Уникальная» реализация, ориентированная только на пополнение словаря в ИГО-формате, вряд ли целесообразна. Желательно иметь возможность не слишком сложной настройки предлагаемого инструментария для сопровождения других электронных словарей.

7. Программная реализация инструментов должна выполняться как в локальной (для использования на отдельном компьютере), так и в распределенной (для работы в сетях Интернет или Интранет) версиях.

Следствием сформулированных выше требований является качество, очень важное для применения подобного инструментального комплекса в условиях вуза: существенно снижается уровень компетенции и специализации, необходимой для работы по дальнейшему развитию ИГО-словаря, что позволяет привлекать студентов, в том числе и младших курсов, в качестве ее исполнителей. При этом студенты довольно быстро осваивают эти инструменты и приступают к работе со словарем.

И наконец, последнее. Мы здесь обсуждаем только задачу добавления словарной статьи, поскольку изменение уже имеющейся в словаре статьи выполняется как две последовательные операции: удаление ее прежнего варианта и добавление затем нового, измененного.

Архитектура инструментального комплекса. «Производственный» цикл пополнения словаря

На рис. 1 представлены основные компоненты комплекса программных инструментов для сопровождения электронного словаря. Такой словарь мы будем далее считать *целевым*.

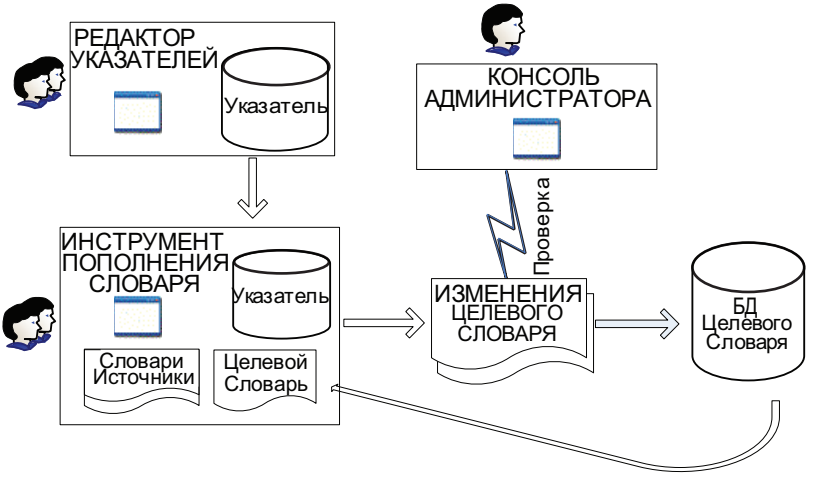


Рис. 1. Основные компоненты комплекса программных инструментов для сопровождения электронных грамматических словарей

Комплекс включает три основных инструментальных приложения, с которыми работают, соответственно статусу, пользователи трех разных категорий:

– *инструмент пополнения словаря*, предоставляющий возможность пользователям – составителям словаря формировать различными способами словарные статьи;
– *редактор указателей*, используя который специалисты с достаточным лингвистическим образованием могут корректировать и пополнять новыми правилами указатели для сопоставления ТФ вводимым в словарь основам;
– *консоль администратора*, предоставляющая *администратору* проекта по развитию словаря, как наиболее квалифицированному и ответственному специалисту, осуществлять функции по контролю и учету изменений словаря.

В распределенной версии инструментального комплекса эти три компонента являются клиентскими приложениями.

Большую часть работы по заполнению словаря выполняют его составители. Поэтому компонент для пополнения словаря можно считать «центральным» в комплексе инструментов. Его функциональные возможности облегчают, насколько это возможно, разработку новых словарных статей.

Прежде всего, составитель имеет возможность проверить, представлено ли данное слово в актуальной текущей версии целевого словаря или нет, а если да, то проверить правильность такого представления.

При формировании новой словарной статьи составитель может использовать различные вспомогательные средства. Одним из них является обращение к источникам.

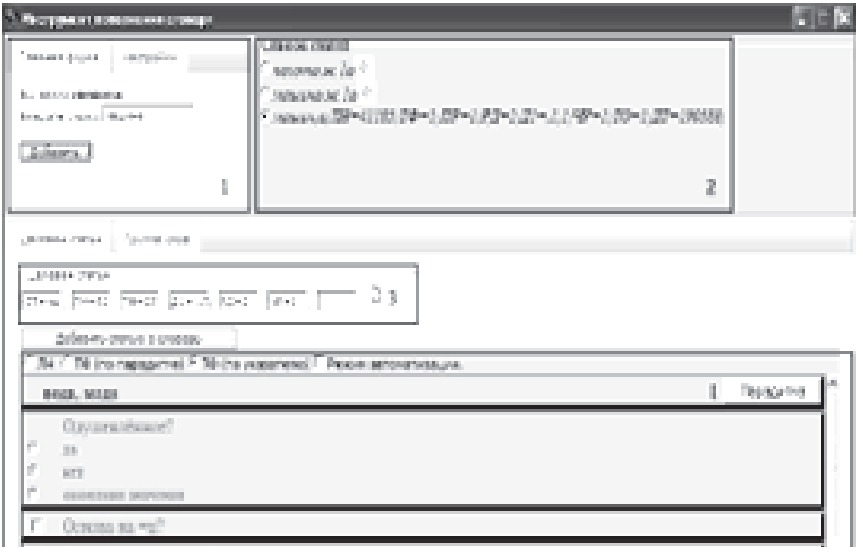


Рис. 2. Окно инструмента для пополнения ИГО-словаря
1 – главная форма приложения;
2 – список статей, найденных по заданным словам в источнике и целевом словаре;
3 – текущее состояние разрабатываемой статьи в формате целевого словаря;
4 – «стековый мастер» – интерфейс указателя для составителей словаря (чем ниже страница мастера, показывающая правило, тем раньше оно было рассмотрено пользователем)

Настоящая, применяемая на практике, версия комплекса инструментов ориентирована на пополнение ИГО-словаря как целевого. В роли источников в ней выступают «Грамматический словарь русского языка» А.А. Зализняка в электронном виде и сам целевой ИГО-словарь. Если вводимое в ИГО-словарь слово содержится в словаре А.А. Зализняка, то часть параметров ИГО его основы можно вычислить по описывающей это слово статье словаря А.А. Зализняка и такое преобразование реализовано в компоненте пополнения словаря. Если в ИГО-словаре найдена лексема, аналогичная новому слову, то значения части параметров ИГО-основ такой лексемы могут быть перенесены в разрабатываемую статью. Рассматриваемый инструмент дает и такую возможность.

В ряде случаев номер ТФ для новой основы также может быть определен по аналогии при обращении к словарям-источникам. В других случаях составитель может воспользоваться актуальной версией соответствующего указателя.

Компонент пополнения словаря не дает составителю возможности непосредственно внести свои изменения в целевой словарь. В процессе работы составителя лишь формируется *пул изменений* целевого словаря как совокупность новых и/или измененных статей.

Периодически порции таким образом подготовленных новых словарных статей составитель передает администратору. Последний, используя консоль администратора, проверяет полученные от составителя словарные статьи, делает необходимые исправления и фиксирует изменения в БД целевого словаря. Эта БД, по существу, представляет журнал изменений целевого словаря. Актуальные состояния целевого словаря (после внесения изменений в БД) выгружаются в форматы его представлений, с которыми работают использующие его приложения (среди них и инструмент пополнения этого словаря).

Кратко охарактеризуем организацию обмена данными между различными специалистами, разрабатывающими словарь. При использовании ими локальной версии рассматриваемого программного комплекса возможен только обмен файлами. Распределенная версия комплекса инструментов предполагает, что БД целевого словаря локально поддерживается на сервере. Таким образом, для этой БД снимается техническая проблема синхронизации параллельно разрабатываемых версий[12]. Однако другие информационные ресурсы и в этой версии удобно передавать для использования на стороне клиента. Техническая проблема синхронизации «параллельно» модифицируемых версий одних и тех же информационных ресурсов в настоящее время разрабатывается другим исполнителем в Учебно-научном центре программного и лингвистического обеспечения интеллектуальных систем (УНЦ ПиЛОИС) РГГУ.

Средства реализации

- Выбор средств реализации обсуждаемого здесь программного комплекса в первую очередь обусловлен следующими причинами.
- Реализация продукта проводится в настоящее время в виде локальной и предполагается в дальнейшем в виде распределенной версии. Желательно, чтобы при этом, насколько это возможно, использовались одни и те же средства реализации.
- В силу ряда причин реализация пока будет проводиться на основе платформы Windows.
- Существует возможность применения разработанных ранее в УНЦ ПиЛОИС средств и технологий, упрощающих реализацию специфицированного программного продукта.

В итоге для разработки инструментального комплекса в целом используются следующие технологии и средства.

1. Технология DHTML (Dynamic HTML) с использованием языка сценариев JavaScript и библиотека JavaScriptMVC Framework[13] для реализации всех трех основных инструментов, что позволит легко оформить их как клиентские приложения в будущей распределенной версии. Браузер Internet Explorer версии 6.0 и выше применяется в качестве среды выполнения для поддерживающих интерфейс с пользователем приложений. Используются также HTML-компоненты (HTC)[14].

2. Применение специально разработанных несложных предметных XML-языков для обмена данными между различными приложениями и их составными частями. Использование XSLT для отображения XML-документов в их HTML-представления с целью визуализации и редактирования.

3. БД целевого словаря и БД для каждого из указателей реализованы в интегрированной среде разработки (IDE) MS Access. Такой выбор делает удобным макетирование БД как информационных ресурсов системы в целом. Этот формат приемлем и для обмена файлами БД между разработчиками словаря (MS Office, включающий Access, имеется на каждом рабочем и домашнем компьютерах). Небольшое количество участников проекта по расширению ИГО-словаря дает возможность даже поддерживать эти БД в формате MDB на сервере в случае распределенной версии. При необходимости несложно конвертировать эти БД в формат MS SQL Server или в формат популярных баз данных, работающих на платформе Linux (например, MySQL) при переходе на Unix-сервер.

4. Применение объектной библиотеки ADO (ActiveX Data Objects) в качестве API к указанным выше БД.

Некоторые особенности реализации комплекса программных инструментов для сопровождения электронного ИГО-словаря

Словари различаются содержанием и форматом представления статей. В то же время желательно, чтобы один из самых важных инструментов – редактор словарных статей – был структурно ориентирован на их представление, показывая их как можно нагляднее. Ясно, что интерфейс такого редактора будет уникальным. Как будет показано ниже, сами задачи, возникающие при сопровождении машинных словарей, используемых различными программными системами, могут отличаться от того, что приходится делать при составлении (или ведении) словарей, ориентированных на пользователя-человека, в том числе словарей печатных. Таким образом, подобные инструментальные системы разрабатываются индивидуально для каждого словаря. Однако наличие в них компонент со схожей функциональностью и некоторое подобие в архитектуре позволяют искать упрощающие их создание технологии.

Подобная технология[15] была предложена М.Е. Епифановым в 2006 г. С тех пор она развивается и применяется на практике[16] автором этой статьи. На ее основе разработаны два комплекса инструментальных средств: один, рассматриваемый в настоящей работе, и другой[17], предназначенный для сопровождения печатного латинско-русского словообразовательного словаря[18] (первая версия инструментов выполнена Р.А. Веретеновым в рамках дипломного проекта, с тех пор поддерживается и развивается автором данной статьи).

Кратко охарактеризуем применение этой технологии в реализации рассматриваемых программных инструментов.

Все они разработаны на основе применения открытой (open source) библиотеки JavaScriptMVC Framework, удобной для построения *богатых клиентов* (Rich Internet Application – «богатые Интернет-приложения», работающего на клиентской, в смысле архитектуры «клиент – сервер», стороне). Такие приложения обладают функциональностью традиционных настольных приложений и не поддерживаются браузерами непосредственно. Инструменты пополнения ИГО-словаря являются «богатыми клиентами».

Основными достоинствами JavaScriptMVC являются: поддержка архитектуры Model – View – Controller, предоставляемые средства тестирования и развертки[19].

Model – View – Controller (MVC, «Модель – представление – поведение» или «Модель – представление – контроллер») – архитектура программного обеспечения, в которой модель данных приложе-

ния, пользовательский интерфейс и управляющая логика разделены на три отдельных компонента так, что модификация одного из них оказывает минимальное воздействие на другие[20].

Как уже говорилось выше, одной из возможностей автоматизации пополнения целевого электронного словаря является привлечение «сторонних» словарей-источников, по которым автоматизированно или полностью автоматически можно строить требуемые данные. Например, по статье словаря А.А. Зализняка можно построить часть ИГО-основы для того же самого слова или слова с такой же парадигмой. При этом возникает необходимость преобразования данных из формата одного словаря в формат другого.

Предлагаемая система сопровождения электронных словарей реализована как открытая для добавления новых словарей в качестве источников и/или целевых. Но при добавлении словаря приходится заново решать задачи парсирования (разбора) формата представления словарной статьи, отображения формата представления в формат визуализации на форме инструмента пополнения словаря, конверсии словарной статьи из формата одного словаря в формат другого. Уникальная реализация этих трех задач достаточно трудоемка сама по себе, а с увеличением числа используемых словарей затруднит и поддержку программного комплекса в целом.

Применение JavaScriptMVC и разработанной ранее в УНЦ ПиЛОИС объектной библиотеки оболочек источников данных[21] дает возможность унифицировать и существенно ускорить реализацию этих трех задач для инструмента пополнения словаря и для консоли Администратора.

В программной реализации редактора указателей существенным образом используется технология интерактивной визуализации и редактирования иерархических структур с «богатым содержанием узлов»[22]. Указатель – это дерево с большим количеством представляющих его правила узлов. Целиком такое дерево загружать в редактор нецелесообразно. Указанная технология позволяет подгружать фрагменты дерева «на лету», т. е. осуществляется так называемая *ленивая подкачка*: дочерние узлы подгружаются, когда пользователь приблизился к родительскому узлу, раскрывая (в смысле «свернуть – развернуть») узлы, предшествующие ему при обходе, и просматривая и/или редактируя содержащиеся в них правила. Узлы из БД указателя в редактор и обратно передаются в специальном XML-формате. Отображение XML-представления узла указателя в соответствующий HTML-формат для просмотра и редактирования реализовано при помощи XSLT.

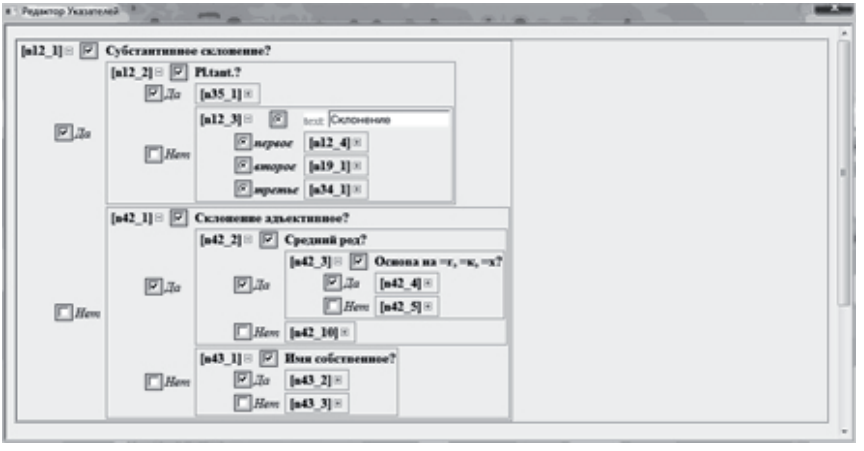


Рис. 3. Общий вид инструмента редактирования указателей

Разработана отдельная утилита импорта указателей, модифицированных другими пользователями, что облегчает распространение новых указателей среди пользователей комплекса инструментов. После импорта указатель становится доступным в стековом мастере инструмента пополнения словаря.

Заключение

Разработан комплекс программных инструментов, облегчающих коррекцию и пополнение новыми словами ИГО-словаря. Положенная в основу его реализации технология допускает его модификации с целью добавления новых словарей-источников или переориентации его для сопровождения другого специализированного электронного грамматического словаря, экономя при этом человеко-временные ресурсы.

Примечания

[1] См.: *Белоногов Г.Г., Калинин Ю.П., Хорошилов А.А.* Компьютерная лингвистика и перспективные информационные технологии. М.: Русский мир, 2004. С. 11.

[2] *Кобзарева Т.Ю.* Иерархия задач поверхностно-синтаксического анализа русского предложения // НТИ. 2007. Сер. 2. № 1. М.: ВИНИТИ, 2007. С. 23–35.

[3] *Баталина А.М., Айриян Г.Ю., Епифанов М.Е., Кобзарева Т.Ю., Кушнарева Е.В., Лахути Д.Г.* Объектная среда для отладки алгоритмов поверхностно-синтак-

сического анализа // Десятая национальная конференция по искусственному интеллекту с международным участием КИИ–2006: Труды конференции. Т. 2. М.: Физматлит, 2006. С. 589–597; *Баталина А.М., Епифанов М.Е., Ивличева О.О., Кобзарева Т.Ю., Лахути Д.Г.* Инструментальная среда для экспериментов с алгоритмами поверхностно-синтаксического анализа // Компьютерная лингвистика и интеллектуальные технологии: Труды Международной конференции «Диалог'2004». М.: Наука, 2004. С. 32–38.

[4] В составе этой группы составлением лингвистических алгоритмов поверхностно-синтаксического анализа начала заниматься Т.Ю. Кобзарева.

[5] См.: *Владимирова Е.В., Карпова Г.Д., Лесскис Г.А., Уриновская И.Д.* Словарь окончаний в системе автоматического индексирования документов «Скобки» // НТИ. 1985. Сер. 2. № 6. С. 29–31; *Пархоменко В.Ф.* Система автоматического индексирования документов СКОБКИ ОС ЕС // Методические материалы и документация по пакетам прикладных программ. Вып. 23. М.: МЦНТИ, 1983.

[6] Там же.

[7] См.: *Карпова Г.Д., Пирогова Ю.К., Кобзарева Т.Ю., Микаэлян Е.В.* Компьютерный синтаксический анализ: описание моделей и разработок. Итоги науки и техники (серия «Вычислительные науки»). Т. 6. М.: ВИНИТИ, 1991.

[8] Пример частеречной омонимии, которая должна сниматься «хорошей» системой синтаксического анализа.

[9] См.: *Зализняк А.А.* Грамматический словарь русского языка. Словоизменение. М.: АСТ-ПРЕСС, 2008. С. 794.

[10] В рассматриваемом словаре лексемы пронумерованы. Все основы, относящиеся к одной и той же лексеме, содержат ее номер в качестве одного из параметров. Поэтому, упорядочив словарь основ по номерам лексем, мы могли бы собрать словарные статьи в блоки размещенных подряд основ для каждой лексемы. Такая утилита реализована и применяется для решения некоторых задач. Имеется утилита, восстанавливающая порядок статей, нужный для морфоанализатора. Однако такое переупорядочивание уже само является некоторым средством автоматизации редактирования словаря основ, требующим удобного интерфейса. Но его применение не снимает другие обсуждаемые здесь трудности редактирования словаря основ «вручную».

[11] Справочная документация по ИГО-словарю, включая указатели для сопоставления ТФ-основам, была также разработана рабочей группой под руководством Г.А. Лесскиса в «Информэлектро» в 1980-е годы.

[12] Не имея здесь возможности обсуждать этот вопрос, заметим, что репликация баз данных, предоставляемая, например, в MS Access, лишь частично решает эту проблему.

[13] JavaScriptMVC – официальный сайт. Главная страница [Электронный ресурс] // JavaScriptMVC – an open source javascript framework. URL: http://1-5.javascriptmvc.com.

[14] HTC Reference. URL: http://msdn.microsoft.com/en-us/library/ms531018%28VS.85%29.aspx.

[15] *Хохряков И.А., Веретенов Р.А., Айриян Г.Ю., Епифанов М.Е.* Об инструментальных системах сопровождения печатных и электронных словарей // Информационное общество. Интеллектуальная обработка информации. Информационные технологии. Труды 7-й международной конференции НТИ'2007 (Москва, 24–26 октября 2007 г.). М.: ВИНИТИ, 2007. С. 361–362.

[16] Там же.

[17] Там же.

[18] На протяжении многих лет разрабатывается Г.В. Петровой «вручную». Мы надеемся, что применение упомянутых здесь инструментов облегчит работу по дальнейшему развитию этого словаря в будущем. (Имеется печатное издание словаря: Латинско-русский словообразовательный словарь: Ок. 20 000 слов / Авт.-сост. Г.В. Петрова. М.: Оникс: Мир и образование, 2008. С. 704).

[19] См.: JavaScriptMVC [Элекронный ресурс] // Сайт open-source проекта Java ScriptMVC. [USA, 2010]. URL: http://www.javascriptmvc.com (дата обращения: 20.12.2010).

[20] См.: *Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж.* Приемы объектно-ориентированного проектирования. Паттерны проектирования. СПб.: Питер, 2001. С. 368.

[21] См.: *Ершова Е.С., Епифанов М.Е.* Графический конструктор структур объектов как интерфейс инструментальной объектной среды // Девятая национальная конференция по искусственному интеллекту с международным участием КИИ 2004: Труды конференции. Т. 2. М.: Физматлит, 2004. С. 498–507.

[22] *Айриян Г.Ю.* Об интерактивной визуализации и представлении иерархических структур в гуманитарных приложениях // Девятая национальная конференция по искусственному интеллекту с международным участием КИИ 2004: Труды конференции. Т. 2. М.: Физматлит, 2004. С. 443–451.

Abstracts

D.A. Larin

STAGES OF CRYPTOGRAPHIC ACTIVITY IN RUSSIA

The given work is the author's first step in this direction, periodization is provided by reference to particular cryptographic ideas which influenced on the national cryptography during certain defined stages. The organizational changes occurring in cryptographic services of our country are considered in some degree. Certainly, it is necessary to continue research in the given direction. So, in cryptographic activity of Russia it is offered to allocate following stages.

Keywords: Russia, cryptography, encoding, cryptoanalysis, information security, cipher, encryptor.

A.E. Baranovich

SEMANTIC ASPECTS OF INFORMATION SAFETY: CONCENTRATION OF KNOWLEDGE

From positions of the information-evolutionary approach the basic directions of intellectual systems information safety are investigated. The main attention in the present work is concentrated on a direction of their protection "from the information". Article continues a cycle of works devoted to semantic-pragmatical aspects of information safety maintenance.

Keywords: axiology, knowledge, knowledge concentration, intellectual systems, information redundancy, information safety, semantics, semantic filters, teleology, safety threats.

A.E. Satunina, L.A. Sysoeva

ANALYSIS OF TRANSITION MODELS TO SERVICE-ORIENTED ARCHITECTURE IN UNIVERSITY INFORMATION SYSTEM

The key starting points and scenarios for the transition to service-oriented architecture (SOA) information system are described. The benefits of SOA information system university are justified. Finally, the model of university information system based on SOA is provided.

Keywords: service oriented architecture (SOA), information system architecture.

A.S. Sysoev

METADATA SERVICES DEVELOPMENT
FOR THEIR INFORMATION SECURITY ASSESSEMENT

The information security in information systems with service-oriented architecture (SOA) is considered. To define the IT-services security level the method of forming the metadata services system based on domain decomposition is used. The parameters of IT service portfolio are described.

Keywords: service-oriented architecture (SOA), information system architecture, information security.

M.I. Zabezhaylo

TOWARDS MODERNIZATION OF THE INFORMATION
SYSTEMS COMPLEX IN LARGE COMMERCIAL BANK

The technology of formalization and problems solving in projects of IT Infrastructure modernization in Russian commercial banks is offered. Its possibilities and restrictions are discussed.

Keywords: bank business, modernization, IT Infrastructure, formalization methods, bank IT projects.

P.V. Pekichev

THEORETIC ESTIMATION OF COVERT CHANNELS
CAPACITY

The purpose of this article is to review the method of estimating the information amount that can be passed through a hidden channel, based on the capacity calculation of data transmission with noise. It focuses on the fundamental theorem of Shannon's channel with noise. In this paper the theorem is proved for a special kind of noise generating by regular users of a safety network.

Keywords: hidden data transmission channel, Shannon's theorem, capacity, random noise, equivocation.

A.I. Svintsitsky

SOME STATISTICAL COVERT CHANNELS

Covert channels are essential threat of information security processed in protected network segments because of vulnerability

and superfluous of modern used network protocols. Moreover, problems of covert information transmission, creation, analysis and protection from covert channels were left out during network protocols development. In this paper, one of possible statistical covert channels classes is considered for the purpose of effective algorithms creation in order to provide detection and protection from these types of channels.

Keywords: covert channels, statistical covert channels, network protocols, information security.

A.A. Grusho, N.A. Grusho, E.E. Timonina

ARTIFICIAL UNRELIABILITY OF THE INFORMATION AS MEAN OF ITS PROTECTION

In this paper the main principles of unreliable information usage for the protection organization are considered. It is shown that by means of unreliability the problems of confidentiality, integrity and availability protection can be solved. The basic methods of unreliability injection and efficiency estimation of unreliable information usage methods are considered.

Keywords: information security, information unreliability, intellectual noise.

S.M. Iglitskaya

TOWARDS STRUCTURALLY-ALGEBRAIC AND SEMANTIC-PRAGMATICAL ANALYSIS OF MUSICAL TEXT

In the article certain questions of musical text research in terms of musical and verbal communication channel efficiency estimation are considered. Basic sections are devoted to comparative analysis of null and first approximation in John von Neumann discrete message model of verbal text and musical strict styled text. Also some questions of musical text semantics are mentioned.

Keywords: musical text, verbal text, communication channel, John von Neumann discrete message model, final alphabet, text semantics.

A.S. Malkova

FORMAL PROBLEM SOLVING OF ESTIMATIONS DISCREPANCY IN VALUABLE STATEMENTS (ON THE MATERIAL OF RUSSIAN PROVERBS)

In the paper the proverb is considered as a particular semantic item corresponding to specific (axiology-oriented) data in real world. This determines the formal semantic model of proverb text. In the paper we examine the regular occurrence of contradictory judgment and the ways of its settlement in the model.

Keywords: knowledge representation, axiology-oriented statement, proverb.

Y.K. Sergeev

ANALYSIS OF SECURITY THREATS IN VIRTUAL INFORMATION SYSTEMS

In given article the author considers the core vulnerabilities of virtualization systems taking into account, so-called, "natural" vulnerabilities, resulting from programmers errors during the virtualization tool design. The vulnerabilities relation with possible ways of attacks realization and necessary conditions for these attacks is described. Finally, this relation derives security threats of the information processed in virtual information systems.

Keywords: virtualization, information security, threats, vulnerabilities, intruder, attacks.

R.R. Gilyazov

TIME ESTIMATION BETWEEN TWO PASSED EVENTS IN THE OPERATING SYSTEM

In an operating system functioning cycle many events occur at random moments of time. The problem of time calculation which has passed between two events is of concern. Especially if one of them causes another one. The class of such problems includes such themes as, for example, the time estimation of the protective software reaction on harmful influence, the operating system influence on operating time of some machine instructions sequences, the profiling, etc. However because of external factors influence the time calculation is a nontrivial task.

Keywords: time estimation, information security, operating system influence, heuristics, protective software reaction time, timers.

V.V. Chernyakovsky

METHOD FOR DYNAMIC FINDING OF GLOBAL KERNEL VARIABLES IN WINDOWS NT OPERATING SYSTEMS

In this paper we present a new method for dynamic finding of global kernel variables. The theoretical basics of its creation possibility and the experimental confirmation for its practical usage are provided.

Keywords: operating system, Windows NT platform, global kernel variables, kernel debugger, driver, rootkit, security software.

I.G. Kazovsky

PLATFORM FOR CONSTRUCTION OF THE RULE-BASED MACHINE TRAINING ALGORITHMS

The purpose of this article is to describe the program, which is a universal platform for developing, debugging and comparing rule-based machine learning algorithms. This article adresses to common issues of platform's organisation, limitations of algorithms, that can be executed on this platform. Also, the author gives an example of education using JSM-method on two different data sets.

Keywords: platform, rule-based machine learning algorithms, system-independence, data representation invariance, algorithm invariance.

I.A. Khokhryakov

IMPLEMENTATION OF PROGRAM TOOLS FOR THE RUSSIAN ELECTRONIC GRAMMATICAL DICTIONARIES SUPPORT

The article describes the implementation characteristics of the applications facilitating correction of and input of new words into specialized electronic grammatical dictionaries. The dictionary described in this article is intended primarily for syntactic analysis of Russian texts, but it can also be applied for other purposes. The primary criterion on which the described realization is based is the minimization of time-manpower resources needed for these instruments development and for their consequent modification, especially for their adjustment to other electronic dictionaries formats support.

Keywords: natural-language Russian text processing, electronic dictionaries for inflexional languages, dictionary formats, syntactic text analysis, software implementation, local and distributed software architecture, relational DB, problem-oriented XML languages, scenario languages.

Сведения об авторах

Баранович Андрей Евгеньевич – доктор технических наук, профессор кафедры компьютерной безопасности Института информационных наук и технологий безопасности при Российском государственном гуманитарном университете (ИИНиТБ РГГУ), barae@rambler.ru

Гилязов Руслан Раджабович – студент, Московский государственный университет им. Ломоносова, факультет вычислительной математики и кибернетики, irusrubin@gmail.com

Грушо Александр Александрович – доктор физико-математических наук, профессор, зав. кафедрой компьютерной безопасности ИИНиТБ РГГУ, grusho@yandex.ru

Грушо Николай Александрович – преподаватель кафедры компьютерной безопасности ИИНиТБ РГГУ, info@itake.ru

Забежайло Михаил Иванович – кандидат физико-математических наук, zmivan@gmail.com

Иглицкая Софья Михайловна – аспирантка кафедры общей информатики ИИНиТБ РГГУ, sofa.sofa@mail.ru

Казовский Илья Григорьевич – аспирант кафедры математики, логики и интеллектуальных систем в гуманитарной сфере РГГУ, gkazovsky@gmail.com

Ларин Дмитрий Александрович – кандидат технических наук, доцент ИКСИ, greattzar@yandex.ru

Малкова Анастасия Сергеевна – соискатель РГГУ, asmalkova@gmail.com.

Пекичев Павел Валентинович – аспирант Московского государственного университета им. М.В. Ломоносова, pavelpek@mail.ru

Сатунина Анна Евгеньевна – кандидат экономических наук, ведущий научный сотрудник, декан факультета информатики ИИНиТБ РГГУ, aesat@mail.ru

Свинцицкий Антон Игоревич – аспирант факультета вычислительной математики и кибернетики Московского государственного университета им. М.В. Ломоносова, anton.svintsitskii@gmail.com

Сергеев Юрий Константинович – аспирант кафедры компьютерной безопасности ИИНиТБ РГГУ, ysergeev@gmail.com

Сысоев Александр Сергеевич – аспирант факультета информатики ИИНиТБ РГГУ, zt0@mail.ru

Сысоева Леда Аркадьевна – кандидат технических наук, доцент, директор Центра дистанционных технологий обучения РГГУ, leda@rggu.ru

Тимонина Елена Евгеньевна – доктор технических наук, доцент, профессор кафедры компьютерной безопасности ИИНиТБ РГГУ, eltimon@yandex.ru

Хохряков Игорь Александрович – программист УНЦ программного и лингвистического обеспечения интеллектуальных систем РГГУ, igorkhokh@yandex.ru.

Черняковский Валерий Валерьевич – аспирант кафедры компьютерной безопасности ИИНиТБ РГГУ, v.chernyakovskiy@gmail.com

Information about the authors

Baranovich Andrew E. – doctor of engineering science, professor of computer security department in Institute for Information Sciences and Security Technologies of Russian State University for the Humanities (IISaST of RSUH), barae@rambler.ru

Chernyakovskiy Valeriy V. – postgraduate of computer security department in IISaST of RSUH, v.chernyakovskiy@gmail.com

Gilyazov Ruslan R. – student, Moscow State University, faculty of Computational Mathematics and Cybernetics, irusrubin@gmail.com

Grusho Alexander A. – doctor of physico-mathematical science, professor, head of computer security department in IISaST of RSUH, grusho@yandex.ru

Grusho Nickolay A. – lecturer of computer security department in IISaST of RSUH, info@itake.ru

Iglitskaya Sofya M. – postgraduate student of computer science department of IISaST of RSUH, sofa.sofa@mail.ru

Kazovsky Ilya G. – postgraduate student, RSUH, gkazovsky@gmail.com

Khokhryakov Igor A. – programmer of Educational and scientific center for software and linguistic support of intelligent systems in RSUH, igorkhokh@yandex.ru

Larin Dmitry A. – candidate of engineering science, associate professor of IKSI, greattzar@yandex.ru

Malkova Anastasia S. – postgraduate of RSUH, asmalkova@gmail.com

Pekichev Pavel V. – postgraduate of Moscow State University, faculty of Computational Mathematics and Cybernetics, pavelpek@mail.ru

Satunina Anna E. – candidate of economics, leading research associate, dean of computer science faculty in IISaST of RSUH, aesat@mail.ru

Sergeev Yuri K. – postgraduate of computer security department in IISaST of RSUH, ysergeev@gmail.com

Svintsitskiy Anton I. – postgraduate of Moscow State University, faculty of Computational Mathematics and Cybernetics, anton.svintsitskii@gmail.com

Sysoev Alexander S. – postgraduate of computer science faculty in IISaST of RSUH, zt0@mail.ru

Sysoeva Leda A. – candidate of engineering science, associate professor, director of Distance learning technologies center of RSUH, leda@rggu.ru

Timonina Helen E. – doctor of engineering science, professor of computer security department in IISaST of RSUH, eltimon@yandex.ru

Zabezhaylo Michael I. – candidate of physical and mathematical sciences, zmivan@gmeil.com

Заведующая редакцией *Е.Е. Жигарина*

Художник номера *В.Н. Хотеев*

Корректор *Н.К. Егорова*

Компьютерная верстка *Н.В. Москвина*

Формат 60×90 1/16.
Усл. печ. л. 15,0. Уч.-изд. л. 15,2.
Тираж 1050 экз. Заказ № 209

Издательский центр
Российского государственного
гуманитарного университета
129663, Москва, Миусская пл., 6
www.rggu.ru
www.knigirggu.ru