Российский государственный гуманитарный университет
Russian State University for the Humanities

RSUH/RGGU BULLETIN
№ 11 (133)

Academic Journal

Series:
Computer Science. Data Protection. Mathematics

Moscow 2014

ВЕСТНИК РГГУ
№ 11 (133)

Научный журнал

Серия
«Информатика. Защита информации. Математика»

Москва 2014

УДК 94 (560)
ББК 63.3(5)я5

Номер подготовили:
А.А. Тарасов
Е.И. Познякова

ISSN 1998-6769

СОДЕРЖАНИЕ

Технологии

CONTENTS

Technology

От редакции

Предлагаем вашему вниманию издание серии «Информатика. Защита информации. Математика» журнала «Вестник РГГУ», посвященное таким проблемам, как: анализ рисков и угроз информационной безопасности, проектирование систем защиты, функциональная безопасность, семантические аспекты информатики и др.

В данном выпуске в качестве темы номера выбран анализ рисков информационной безопасности. Несмотря на то что этой проблеме посвящено много научных работ, разработан ряд международных стандартов, до настоящего времени актуальны проблемы разработки методик анализа рисков. Анализ рисков позволяет сделать безопасность экономически эффективной, актуальной, своевременной и способной реагировать на угрозы, категоризовать угрозы, определить и обосновать оптимальную стоимость защитных мер.

Приглашаем авторов – преподавателей РГГУ и его филиалов, сотрудников научных центров, представителей большого и малого бизнеса, аспирантов, докторантов – для публикации результатов научных исследований по современной проблематике информационных технологий и математики.

Материалы для журнала просим оформлять в соответствии с принятыми нормами, установленными ВАК для рецензируемых научных изданий, и направлять их электронной почтой по адресу: vestnik@rggu.ru на имя ответственного редактора серии А.А. Тарасова.

С.А. Желтов

ОБЩАЯ МЕТОДИКА ПРАКТИЧЕСКОЙ АПРОБАЦИИ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ В ОТНОШЕНИИ РИСКОВ РЕАЛИЗАЦИИ УГРОЗ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ПРИ ИХ ИСПОЛЬЗОВАНИИ*

Статья посвящена некоторым аспектам оценки технических (вычислительных) возможностей потенциального нарушителя информационной безопасности автоматизированных систем.

Ключевые слова: информационная безопасность, угрозы и риски безопасности, автоматизированные системы.

На сегодняшний день информационная безопасность (ИБ) есть составная часть предметной области информационно-коммуникационных технологий, динамика развития которой характеризуется беспрецедентно высокими темпами роста. До недавнего времени рост основных показателей вычислительной техники характеризовался законом Мура[1]. В настоящий момент многие эксперты в области вычислительных технологий считают, что развитие отрасли идет опережающими закон Мура темпами, т. е. увеличение происходит быстрее, чем за два года, а именно за полтора. Рост производительности, появление новых архитектур высокопроизводительных вычислительных устройств порождают новые угрозы безопасности информационных систем, характеризуемые снижением фактических временных затрат на преодоление практической стойкости подсистемы защиты до неприемлемого уровня.

Для создания полноценной системы защиты информации, обрабатываемой в коммерческих автоматизированных системах (АС) наряду с разработкой политик безопасности, построения формальных моделей разграничения доступа, использования криптографи-

* Автор выражает глубокую благодарность проф. А.Е. Барановичу за ценные рекомендации и помощь при проведении исследований.

ческих средств защиты, необходимо решать вопросы, касающиеся исследования и оценки технических возможностей потенциального нарушителя ИБ АС.

Новые угрозы информационной безопасности

Основой определения угроз для конкретной информационной системы является выявление потенциальных нарушителей, прогнозирование их возможностей, намерений и тактики действий. Применение существующих методик оценки возможностей потенциального нарушителя не учитывает современные факторы, позволяющие достичь значительного увеличения вычислительной мощности доступных технических средств, что приводит к неадекватному отображению реальных условий функционирования объекта защиты с точки зрения возможных противоправных действий в его отношении. При этом могут неверно оцениваться и действительные риски реализации новых угроз безопасности АС. Это приводит к необходимости своевременного пересмотра требований к средствам защиты в зависимости от темпов роста основных показателей вычислительной техники и появления принципиально новых технических решений. Риски реализации новых угроз безопасности, связанные с вышеперечисленными обстоятельствами, можно оценить:

- с помощью теоретической оценки вычислительной сложности алгоритмов, адаптированных к определенной вычислительной архитектуре;
- с помощью проведения вычислительных экспериментов, позволяющих прагматически охарактеризовать ресурсы, необходимые и достаточные для вскрытия механизма защиты конкретной информационной системы.

Оба способа взаимно дополняют друг друга и позволяют уточнить и детализировать риски безопасности, что в свою очередь дает возможность решить задачу безопасного функционирования АС в условиях интенсивного развития информационных технологий.

Последний подход основан на понятии вычислительной стойкости, которая подразумевает невозможность взлома системы защиты наилучшим из возможных алгоритмов атаки в силу неоправданно высоких затрат используемых вычислительных ресурсов. В настоящее время принято считать, что если для вскрытия требуется больше 2^{80} операций, то взлом становится слишком дорогостоящим и нецелесообразным[2].

При этом необходимо учитывать доступность требуемых для преодоления защиты вычислительных мощностей потенциальному нарушителю. Последнее определяется тремя основными характеристиками:

- удельной производительностью вычислительных устройств – отношение теоретической производительности к энергопотреблению;
- удельным энергопотреблением – отношение удельной производительности к энергопотреблению;
- стоимостью аппаратного обеспечения.

Последняя в большей степени влияет на доступность. Вообще говоря, более правильно рассматривать не абсолютные значения цены вычислительных узлов, а стоимость одного Гфлопса, или экономическую эффективность, т. е. отношение вычислительной мощности к стоимости оборудования.

Примером преодоления систем защиты за счет увеличения вычислительных мощностей является использование графических процессоров. Скорость подбора MD5 – паролей на nVidia GTX580 составляет до 15 800 млн комбинаций в сек., что позволяет найти средний по сложности пароль длиной восемь символов всего за 9 мин.[3]

Методика практической апробации вычислительных систем в отношении рисков реализации угроз информационной безопасности при их использовании

Определение вычислительной стойкости на практике связано с рядом трудностей. Во-первых, невозможно определить понятие наилучшего алгоритма взлома. Во-вторых, реализация известных алгоритмов на новых вычислительных системах после соответствующей адаптации может дать существенный выигрыш по скорости вычислений[4]. Другими словами, алгоритмы, которые считались недостаточно эффективными в рамках классической модели вычислений А. Тьюринга–Дж. Неймана при задействовании новых типов вычислителей, в том числе и мобильных, и (или) способов организации вычислений, могут существенно снизить фактические временные затраты до неприемлемого уровня[5]. Последнее обстоятельство требует более формального подхода к оценке вычислительных возможностей при использовании различных вычислительных систем с процессорами разных типов и архитектур. С учетом всех вышеперечисленных факторов можно предложить

общую методику практической апробации вычислительных систем в отношении рисков реализации угроз ИБ при их использовании:

1. Исследовать программно-аппаратные особенности архитектуры апробируемой вычислительной системы.

2. Исходя из особенностей организации вычислений, проанализировать известные алгоритмы решения вполне определенной задачи на предмет адаптации к используемой архитектуре.

3. Выбрать алгоритм и адаптировать его к рассматриваемой гетерогенной системе.

4. Выполнить теоретическую оценку операционной сложности разработанного (адаптированного) алгоритма, на основании которой сформулировать гипотезу об эффективности возможной реализации.

5. Программно реализовать адаптированный алгоритм.

6. Провести компьютерное моделирование решения задачи.

7. Оценить эффективность использования вычислений на гетерогенной системе с выбранной архитектурой в предложенном алгоритме и реализующей его программе.

8. Проанализировать удельные показатели производительности (вычислительной мощности), энергопотребления используемых вычислителей и их экономической эффективности.

9. По результатам п. 1–8 сформулировать рекомендации и требования к системам защиты в интересах повышения уровня защищенности компонентов, опирающихся на сложность решения исследуемых задач.

Выводы

С учетом динамики развития информационных технологий, массового распространения вычислительных устройств (систем), в том числе и гетерогенного типа, следует ожидать существенного возрастания уровня угроз для коммерческих автоматизированных систем. Более того, при организации массовых вычислений в социально ориентированных информационных ресурсах, основанных на задействовании механизмов «заоблачных» вычислений, следует ожидать существенного возрастания уровня угроз кибербезопасности в отношении систем защиты информации.

Таким образом, в интересах повышения уровня защищенности систем информационной безопасности необходимо учитывать возрастающие угрозы со стороны массового задействования гетерогенных вычислителей, что, в свою очередь, требует организации

непрерывного мониторинга за рынком перспективных вычислительных средств и соответствующей теоретико-экспериментальной апробации новых образцов вычислителей по предложенной методике.

Примечания

[1] Гордон Эрл Мур (3 января 1929, Сан-Франциско, Калифорния) – почетный председатель совета директоров и основатель корпорации Intel, основоположник «закона Мура».

[2] См.: *Смарт Н.* Криптография. М.: Техносфера, 2006.

[3] См.: *Зобнин Е.* Суперкомпьютер из видеокарты: задействуем возможности GPU для ускорения софта [Электронный ресурс] // Сайт журнала «Хакер». URL: http://www.xakep.ru/post/56966/ (дата обращения: 30.04.2014).

[4] См.: *Баранович А.Е.* Введение в предметно-ориентированные анализ, синтез и оптимизацию элементов архитектур потоковых систем обработки данных [Электронный ресурс]. 3-е изд., стереотип., испр. Электрон. дан. [М.: РГГУ, 2010]. 1 электрон. опт. диск (CD-ROM).

[5] См.: *Баранович А.Е., Желтов С.А.* Гетерогенные архитектуры массовых вычислений и новые угрозы кибербезопасности // Системы высокой доступности. 2012. Т. 8. № 2. С. 16–22.

Н.А. Тарасова

СТРУКТУРИЗАЦИЯ ОРГАНИЗАЦИОННЫХ РИСКОВ В СИСТЕМАХ ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ПРЕДПРИЯТИЙ

Рассмотрены содержание и причины возникновения организационных рисков в системах информационной безопасности предприятий. Приведена классификация организационных рисков. В соответствии с существующими методологиями анализа деревьев отказов и факторного анализа информационных рисков проведена структуризация организационных рисков в системах информационной безопасности.

Ключевые слова: система информационной безопасности, организационные риски, структура рисков.

Характерной особенностью современного этапа развития предприятий различных видов является широкое применение современных информационных систем (ИС) для организации протекающих в них бизнес-процессов. При этом актуальной проблемой является обеспечение информационной безопасности предприятия.

Под информационной безопасностью предприятия будем понимать совокупность условий, в которых оно функционирует при воздействии неблагоприятных факторов на информационную инфраструктуру (информационные ресурсы, информационные процессы, программно-технические компоненты, пользователей, обслуживающий персонал, а также поддерживающие системы) без недопустимого снижения качества реализуемых бизнес-процессов.

Для реализации совокупности таких условий предназначена система обеспечения информационной безопасности (СОИБ). СОИБ может недостаточно эффективно реализовывать условия безопасности из-за возможных недостатков в своей организации.

Какие это недостатки и что они могут порождать? Для ответа на данный вопрос сначала рассмотрим понятие «обеспечение».

Оно раскрывается в современной технической литературе двояко – и как вид, и как средство деятельности. Как вид деятельности обеспечение означает совокупность действий, предпринимаемых для того чтобы сделать нечто «вполне возможным, действительным, реально выполнимым», а как средство деятельности – «то, чем обеспечивают кого-нибудь или что-нибудь»[1].

Обеспечение безопасности как вид деятельности есть процесс создания или поддержания безопасных условий функционирования информационной инфраструктуры предприятия, а обеспечение безопасности как средство деятельности – совокупность материальных объектов, людских ресурсов, финансовых, правовых и организационных средств, которые реализуют безопасные условия ее функционирования. Другими словами, СОИБ представляет собой систему деятельности, элементы которой (средства деятельности) реализуют определенный процесс (вид деятельности) поддержания безопасных условий функционирования. При организации такой системы деятельности могут быть недостатки, связанные как со средствами деятельности, так и видом деятельности.

К недостаткам, связанным со средствами деятельности, относятся, например, неоптимальный выбор номенклатуры механизмов обеспечения информационной безопасности, различные проектные ошибки и недочеты при создании СОИБ, нерациональные настройки программно-технических средств защиты информационной инфраструктуры, использование несовершенных методов и средств обеспечения информационной безопасности и др.

К недостаткам второго типа будем относить недостатки, связанные с несовершенством построения и организации функционирования СОИБ. Это несовершенство может проявляться в отсутствии формализованных методик проектирования деятельности по обеспечению информационной безопасности предприятия, в наличии ошибок планирования и проектирования деятельности, в низкой координации работ и регулирования процессов поддержания безопасных условий функционирования, в неправильном подборе и расстановке кадров, в недостаточном участии высшего руководства в обеспечении информационной безопасности, в организационной неготовности предприятия к внедрению СОИБ, в негативных проявлениях человеческого фактора, в частности сопротивлении персонала, психологической усталости, в изъянах политики информационной безопасности предприятия и т. д.

Все эти недостатки могут вызвать события, приводящие к недопустимому снижению качества бизнес-процессов предприятия.

Для анализа таких событий в контексте информационной безопасности предприятия и последствий их наступления в настоящее время широко применяется риск-ориентированный подход. В его рамках нежелательные события связываются с рисками нарушения информационной безопасности предприятия, причем источником этих рисков является сама СОИБ, вернее недостатки ее организации. Данные риски будем называть организационными. Определим их как риски, обусловленные недостатками в организации деятельности по поддержанию безопасных условий функционирования информационной инфраструктуры[2]. При этом под риском будем понимать возможность наступления некоторого неблагоприятного события, влекущего за собой различного рода потери (ущерб). Такие события (функциональные отказы СОИБ) порождаются отклонением от целенаправленной деятельности по поддержанию безопасных условий функционирования информационной инфраструктуры предприятия, которое, в свою очередь, определяется необходимостью решения задачи выбора соответствующих стратегий поведения СОИБ в условиях неопределенности.

Неопределенность вызвана неполнотой или неточностью информации об условиях функционирования информационной инфраструктуры предприятия, в частности угрозах нарушения информационной безопасности и сценариях их реализации; неизвестностью точных значений некоторых параметров сценариев протекания процесса поддержания безопасных условий функционирования, например неполнотой политик безопасности, невозможностью с точностью до 100 % спрогнозировать значение того или иного фактора, непредсказуемостью поведения участников процесса обеспечения безопасных условий функционирования информационной инфраструктуры в ситуации возможного конфликта интересов, например, при децентрализованной схеме администрирования информационной безопасности и др.

Сочетание этих факторов создает в практике организации процесса поддержания безопасных условий функционирования обширный спектр различных видов неопределенностей. Это порождает совокупность различных видов организационных рисков, которые отличаются между собой по месту и времени возникновения, совокупности внешних и внутренних факторов, влияющих на их уровень и, следовательно, по способу их анализа и методам описания.

Эффективное противодействие организационным рискам – многоэтапный процесс, одним из этапов которого является анализ рисков. Анализ может быть качественным или количественным.

Качественный анализ рисков проводится с целью определения их приоритетов. Для этого риски структурируют, ранжируют, оценивают их влияние на бизнес-процессы и оперативность реагирования.

Количественный анализ направлен на получение числовых характеристик штатной реализации бизнес-процессов предприятия.

Количественный анализ, как правило, проводится после качественного анализа в отношении тех рисков, которые выделены в процессе качественного анализа как наиболее значимые.

Одной из первых задачей анализа рисков является их структуризация, хотя в принципе она может решаться еще на этапе идентификации.

В рамках структуризации организационные риски прежде всего классифицируются. Рассмотрим одну из возможных классификаций организационных рисков в СОИБ (рис. 1). При этом основными классификационными признаками будем считать следующие: время возникновения, факторы возникновения, отношение к СОИБ, характер последствий.

По времени возникновения риски подразделяются на ретроспективные, текущие и перспективные риски. Анализ ретроспективных рисков, их характера и способов снижения дает возможность более точно анализировать текущие и прогнозировать перспективные риски.

По факторам возникновения организационные риски могут быть подразделены на сценарные, структурные и параметрические.

Сценарные риски – это риски, обусловленные несовершенством сценариев проведения мероприятий по обеспечению информационной безопасности.

Структурные риски – риски, обусловленные недостатками проектирования структуры СОИБ и ее элементов.

Параметрические риски – риски, обусловленные недостаточностью значений параметров процессов поддержания безопасных условий функционирования информационной инфраструктуры предприятия.

По отношению к СОИБ организационные риски делятся на внешние и внутренние. К внешним рискам относятся риски, непосредственно не связанные с деятельностью в рамках конкретной СОИБ. Это риски доэксплуатационной, проектной природы. К внутренним рискам относятся риски, обусловленные деятельностью в рамках конкретной СОИБ.

По характеру последствий риски подразделяются на допустимые, критические, катастрофические. Данные риски отличаются зонами потерь (ущербов), возникающих при принятии различных по качеству решений по обеспечению информационной безопасности.

Все виды организационных рисков оказывают влияние на деятельность в рамках СОИБ. Данное влияние можно зафиксировать (и в дальнейшем исследовать) при помощи таких структурных конструкций, как деревья отказов (неблагоприятных событий, решений, рисков), а процедура ее исследования содержательно представляет собой метод анализа дерева отказов (Fault Tree Analysis, FTA).

Он описан в отечественных стандартах ГОСТ Р 51901.13-2005 «Менеджмент риска. Анализ дерева неисправностей» и ГОСТ Р 27.302-2009 «Надежность в технике. Анализ дерева неисправностей», а также в ряде зарубежных: NUREG CPH-0492 для атомной энергетики (ориентированная на космос версия этого стандарта используется NASA), стандарте SAE ARP4761 для гражданской аэрокосмической отрасли, MIL-HDBK-338 – для военных систем.

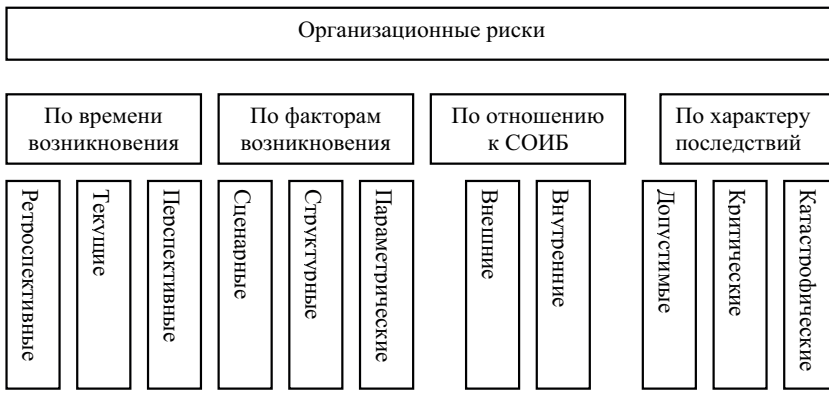


Рис. 1. Классификация организационных рисков

При анализе дерева отказов нежелательные состояния системы – отказы – анализируются с помощью методов булевой алгебры, объединяя последовательность нижестоящих событий (отказов низшего уровня), которые приводят к нежелательному состоянию, интерпретируемому как системный отказ. Дерево отказов (рис. 2) состоит из последовательностей и комбинаций нарушений и неисправностей, и таким образом оно представляет собой много-

уровневую графологическую структуру причинных взаимосвязей, полученных в результате прослеживания опасных ситуаций в обратном порядке, для того чтобы отыскать возможные причины их возникновения.

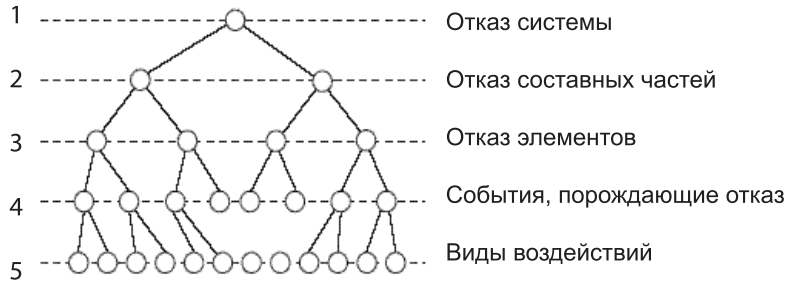


Рис. 2. Условная схема построения дерева отказов

Для построения дерева отказов используются два типа символов, отражающих и логически связывающих некоторые события: символы событий и логические символы (табл. 1)[3].

Анализ дерева отказов эффективно используется в аэрокосмической отрасли, атомной энергетике, химической и перерабатывающих отраслях, в фармацевтической, нефтехимической и других, связанных с высокой степенью риска, чтобы понять, как система может выйти из строя, выявить способ уменьшения рисков или определить вероятность системного отказа.

Деревья отказов содержат структурные элементы, отражающие события, причины наступления которых не исследуются часто по причине отсутствия детальной структуры причинно-следственных связей. Данное обстоятельство порождает задачу разложения рисков на составные части (факторы, обусловливающие риск). Полученная в результате ее решения структура явилась бы хорошим основанием для анализа рисков и реальной базой для объяснения результатов анализа.

Современные подходы к решению этой задачи реализуются в рамках методологии факторного анализа информационных рисков. Риск в соответствии с ней определяется как сочетание частоты наступления неблагоприятного события (Loss Event Frequency, LEF) и возможной величины ущерба (Probable Loss Magnitude, PLM), вызванного его наступлением (рис. 3).

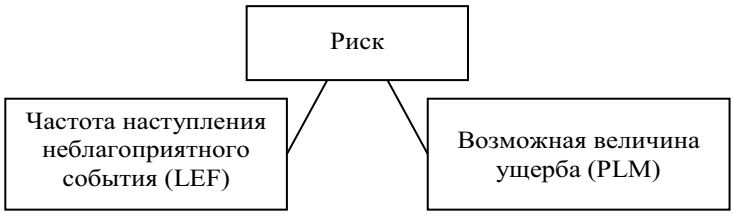
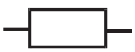


Рис. 3. Основные компоненты риска

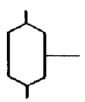
Для того чтобы произошло неблагоприятное событие, необходимо существование некоторой уязвимости в организации СОИБ, связанной с ней угрозы, механизма ее реализации, а также срабатывание этого механизма (оно и вызывает наступление события). Это приводит нас к выделению следующих двух факторов: частоты проявления угрозы неблагоприятного события (Threat Event Frequency) и уязвимости (Vulnerability) (рис. 4).

Таблица 1

Символы, используемые при построении дерева отказов

Символ	Название символа	Сущностная характеристика
Символы событий		
	Событие, вводимое логическим элементом	Возникает в результате взаимодействия событий, происходящих через логическую ячейку
	Первичное, базовое исходное событие	Обеспечено достаточными данными, не требует дальнейшего исследования
	Событие, недостаточно детально разработанное (неразлагаемое первичное событие)	Причины события не исследуются

Окончание таблицы 1

Символ	Название символа	Сущностная характеристика
Логические символы		
	И	Выходное событие происходит, когда имеют место все входные события
	Приоритетное И	Выходное событие происходит, когда все входные события осуществляются в строгом порядке слева направо
	ИЛИ	Выходное событие происходит, если имеются одно или несколько входных событий
	«*m* из *n*» (голосование)	Выходное событие происходит, если поступает не менее *m* из *n* входных событий
	Исключающее ИЛИ	Выходное событие происходит, если случается одно (но не оба) из входных событий
	Запрет	Входное событие вызывает выходное, если происходит условное событие

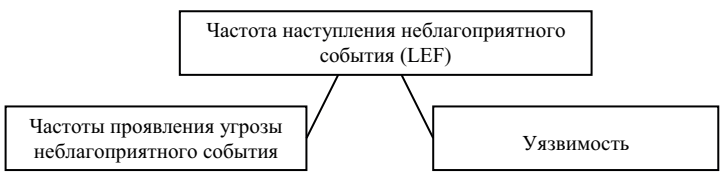


Рис. 4. Факторы частоты наступления неблагоприятного события

Термин «частота проявления угрозы неблагоприятного события» не указывает явно на факт успешного или неуспешного действия субъекта угрозы. Это приводит к необходимости введения двух факторов, которые связаны с частотой проявления угрозы неблагоприятного события: контакт (Contact) и действие (Action) (рис. 5).

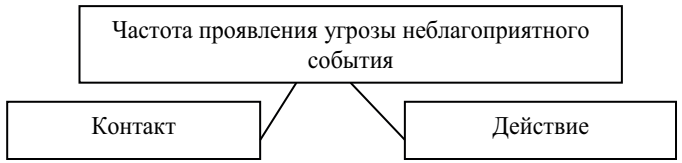


Рис. 5. Факторы частоты проявления угрозы неблагоприятного события

Контакт определяется как вероятная частота на интервале времени, на котором субъект угрозы может войти в контакт с объектом нарушения безопасности.

Контакты могут быть трех видов: случайные, регулярные и преднамеренные. Каждый из этих типов контактов связан с различными факторами.

Действие (воздействие) определяется как вероятность того, что субъект угрозы будет воздействовать на объект нарушения безопасности, когда происходит контакт.

Факторы, определяющие способность противодействовать субъекту угрозы, связаны с уязвимостью – вероятностью того, что объект нарушения безопасности не сможет оказать сопротивления действиям субъекта угрозы – возможность угрозы (Threat Capability) и управление силой (Control Strength)[4].

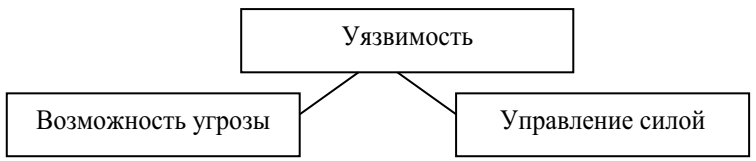


Рис. 6. Факторы, связанные с уязвимостью

Рассмотренные выше факторы определяют частоту наступления неблагоприятного события и являются основой для анализа этого компонента риска.

Анализ другого компонента – возможной величины ущерба – связан с решением достаточно сложной задачи определения допустимых уровней деградации существенных свойств СОИБ при возникновении неблагоприятных состояний (функциональных отказов СОИБ) и ранжирование этих уровней по вкладу в сохранение качества системы. Данная задача включает в себя следующие основные этапы:

– ранжирование функций, выполняемых системой по важности, анализ динамики выполнения функций (время выполнения функций, логика выполнения: цикличность выполнения функций, связь функций между собой, алгоритмы выполнения и т. п.);

– анализ связи элементов СОИБ с выполняемыми функциями;

– формирование системы критериев для оценки критичности функциональных отказов элементов СОИБ;

– разработка соответствующих шкал;

– оценка критичности функциональных отказов элементов СОИБ по выбранным критериям (расчеты, испытания, моделирование, экспертизы и т. п.);

– ранжирование элементов СОИБ по степени критичности их функциональных отказов.

Примечания

[1] См.: *Стрельцов А.А.* Обеспечение информационной безопасности России. Теоретические и методологические основы / Под ред. В.А. Садовничего и В.П. Шерстюка. М.: МЦНМО, 2002. 296 с.

[2] См.: *Тарасова Н.А.* Организационные риски в системе обеспечения информационной безопасности предприятия // Современные проблемы и задачи обеспечения информационной безопасности: Труды Всероссийской научно-практической конференции «СИБ-2014». М.: МФЮА, 2014. С. 73–80.

[3] См.: ГОСТ Р 51901.13-2005 «Менеджмент риска. Анализ дерева неисправностей». М.: Стандартинформ, 2005.

[4] См.: *Jones J.A.* An Introduction to Factor Analysis of Information Risk (FAIR). A framework for understanding, analyzing, and measuring information risk [Электронный ресурс] // CXOWare. URL: http://www.cxoware.com/wp-content/uploads/2013/10/FAIR_Introduction.pdf (дата обращения: 27.01.2014).

Д.А. Ларин

КРИПТОГРАФИЧЕСКАЯ ДЕЯТЕЛЬНОСТЬ В ПЕРИОД КРЫМСКОЙ ВОЙНЫ

160 лет назад, 28 марта 1854 г., Англия и Франция объявили войну Российской империи, таким образом серьезно изменив баланс сил в очередном русско-турецком противостоянии, начавшемся в 1853 г. Причинами войны стало нежелание со стороны европейских держав усиления России, которая практически нанесла поражение Османской империи. На стороне Турции в войну вступили вышеупомянутые державы, а позже к ним присоединилось итальянское королевство Сардиния. Российские войска нанесли коалиции ряд поражений на Кавказе, Камчатке, в районе Санкт-Петербурга, а также Соловецких островов, но, к сожалению, на главном театре военных действий, в Крыму, потерпели поражение[1].

С точки зрения защиты информации эта война стала первой, в ходе которой для управления боевыми действиями были применены принципиально новые средства электросвязи (телеграф). Именно наши военные впервые использовали телеграф для передачи шифрованных сообщений в период боевых действий.

Ключевые слова: Крымская война, криптография, шифрование, дешифрование, шифр, телеграф, связь.

21 октября 1832 г. произошла революция в методах передачи информации. В этот день в Санкт-Петербурге наш соотечественник, выдающийся ученый и изобретатель, глава криптографической службы Российской империи Павел Львович Шиллинг фон Канштадт впервые в мире осуществил передачу сообщения с помощью электричества[2]. В 1836 г. под его руководством была проложена экспериментальная подземная кабельная телеграфная

линия между крайними помещениями здания Адмиралтейства в Санкт-Петербурге, работавшая более года. В том же году Шиллинг предложил подвешивать линейные провода между телеграфными станциями на деревянные опоры. В следующем году Шиллинг начал работу над проектом первой подводной телеграфной линии связи между Петергофом и Кронштадтом. Она не была завершена из-за смерти русского изобретателя 25 июля 1837 г. Работы Шиллинга являются одним из этапов работ по созданию и распространению проволочного телеграфа, они оказали большое влияние на развитие этой области науки и техники в других странах. Тем не менее основным аппаратом, используемом на российских (впрочем, и не только российских) линиях связи, стал знаменитый аппарат американца С. Морзе. Преемником и продолжателем работ П.Л. Шиллинга по развитию и внедрению телеграфа в России стал академик Санкт-Петербургской академии наук Б.С. Якоби. В 1841 г. он построил телеграфную линию между Зимним дворцом и Главным штабом в Санкт-Петербурге, оборудованную оригинальными пишущими аппаратами его конструкции. В 1842 г. подобная линия была проложена от Зимнего дворца до Главного управления путей сообщения и публичных зданий в Санкт-Петербурге. В следующем году была проложена новая линия до дворца в Царском Селе. Построенные Якоби телеграфные линии представляли собой зарытые в землю изолированные медные провода. В 1850 г. Якоби придумал буквопечатающий телеграфный аппарат собственной конструкции. Как говорил об этом устройстве сам ученый, в нем «регистрация знаков осуществлялась с помощью типографского шрифта»[3]. Интенсивная работа по созданию телеграфа в России и связанные с этим теоретические и экспериментальные исследования дают право считать П.Л. Шиллинга и Б.С. Якоби основоположниками телеграфной связи в России.

В XIX в. по мере укрепления российской государственности курьерская связь уже не могла полностью удовлетворить потребности управления страной и вооруженными силами. Сообщения требовалось передавать быстро, а курьерская связь была относительно медленной. Военные конфликты приобретали все большие и большие масштабы, расширялись пространства, на которых одновременно действовали крупные массы войск. Войсками нужно было эффективно управлять. Большое время для передачи конфиденциальных сообщений приводило к несогласованности действий соединений, и даже к их гибели. Русские ученые, инженеры работали над созданием принципиально новых средств передачи информации на дальние расстояния. Это направление исследований было

настолько важным, что изобретения делались порой независимо в России и за рубежом примерно в одно и то же время. Строительство и ввод в эксплуатацию первых линий связи положили начало бурному развитию сети государственного телеграфа. К концу 1855 г. протяженность телеграфных линий в России составила более 5 тыс. км. Первая большая телеграфная линия в 655 км соединила в 1852 г. Санкт-Петербург и Москву. Увеличение количества линий связи приводило к необходимости разрабатывать новые шифры и коды, удобные для закрытия секретной информации, передаваемой с помощью телеграфа.

Основными шифрами, использовавшимися в России во второй половине XIX в., были биграммные шифры П.Л. Шиллинга и биклавные шифры барона Дризена, сменившего Павла Львовича на посту главы криптографической службы России. К усовершенствованию этих шифров и построению ключей для них привлекались специалисты шифровальной службы. Эти шифры применялись в основном в МИД. Также широко использовались коды. Кодовые таблицы объемом до 1000–1200 словарных величин было принято называть словарными ключами и в зависимости от словаря конкретного кода называть французскими, русскими, немецкими. Их применяли в Военном министерстве и МВД, в МИД и некоторых других гражданских ведомствах. Ведущими специалистами по кодам того времени были начальник Цифирного отделения МИД барон Дризен и сотрудник отделения М. Сухотин[4].

После того как электрический телеграф начал использоваться в системе управления государством, большое влияние на его дальнейшее развитие стали оказывать требования военно-стратегического характера. Начавшаяся Крымская война (1853–1856) ускорила постройку телеграфных линий, так как нужна была оперативная связь между крупными городами. В 1854 г. была введена в эксплуатацию телеграфная линия Санкт-Петербург – Варшава протяженностью более 1000 км. В мае 1855 г. закончилось строительство линии Киев – Кременчуг – Николаев – Одесса, а в сентябре того же года была открыта телеграфная связь на новой линии Николаев – Перекоп – Симферополь. С этого времени столица Российской империи получила телеграфную связь с Симферополем, используя для данной цели ранее построенные линии Санкт-Петербург – Москва и Москва – Киев[5]. Таким образом, на некоторое время удалось обеспечить театр военных действий телеграфной связью, по которой передавались шифрованные сообщения.

Во второй половине XIX в. большинство шифров Военного министерства представляли собой коды малого (до 1000 словар-

ных величин) объема. Кодовыми обозначениями здесь являлись трех- и четырехзначные числа. Военные шифры этого типа обычно использовались в течение длительного времени, перерабатывался лишь относительно быстро устаревавший словарь, что объясняется, например, изменением географии военных действий и т. п. Словарные ключи были наиболее распространенным типом шифров, использовавшихся в конце XIX в. в военном ведомстве. Их называли «военными ключами».

Примером шифра, используемого во время Крымской войны, может служить русский словарный ключ № 299 на 600 величин, который был разработан бароном Дризеном и введен в действие в апреле 1854 г. Предназначался этот шифр для связи командиров частей Дунайской армии. После окончания войны этот шифр продолжал использоваться. В 1891 г. им были снабжены российские представители в ряде иностранных городов, а также чиновники в некоторых российских городах, в частности этот шифр применялся в Бухаре, Кашгаре, Кульдже, Сеуле, Пекине, Токио, Иркутске, Омске, Ташкенте, Хабаровске, Урге, Чугучаке, Владивостоке. Ключ № 299 был выведен из употребления только в 1901 г. Еще одним шифром, применявшимся в период Крымской войны, был русский словарный ключ № 300, также составлявший 600 словарных величин, его тоже составил барон Дризен. Этот код был введен в действие в 1855 г., изъят из употребления через некоторое время по окончании Крымской войны, в январе 1857 г. Предназначался он для главнокомандующего Южной армией и военных и морских сил в Крыму князя Горчакова. В 1860 г. этот ключ был послан в Вашингтон. Не употреблялся с 1871 г.[6] Примером дипломатического шифра времен Крымской войны может служить французский биграммный телеграфный ключ № 302. Он был составлен в 1855 г. и введен в действие в 1856 г., а выведен из действия в 1867 г. Употреблялся «уполномоченным при Парижском конгрессе»[7] бароном Брунновым для сношений с МИД. Издано было пять экземпляров этого ключа.

Криптографы МИД прикомандировались в подразделения русской армии еще во время войны с Наполеоном. Накануне и во время Крымской войны в армию направлялись штатские чиновники, которые занимались шифрованием секретной переписки наших военачальников, это было необходимо, чтобы засекретить приказы военного командования на тот случай, если корреспонденция вдруг попадет в руки противника. В войсках их называли «чернильными душами». Один из этих людей сыграл большую роль в спасении русской сухопутной армии в битве на реке Альме 8 сентября 1854 г.

Альминское сражение – пример ошибок и просчетов командовавшего нашей крымской армией генерал-адъютанта князя А.С. Меншикова. Главной задачей наших сухопутных войск была защита Севастополя (главного опорного пункта России в Крыму) с суши от англо-французского десанта, который беспрепятственно высадился в Евпатории и двинулся на юго-запад. Русские войска в количестве 35 тыс. человек при 89 орудиях противостояли 60-тысячной армии антироссийской коалиции, имевшей 112 орудий. 7 сентября силы коалиции подошли к реке Альма и начали перестрелку с нашими войсками. Диспозиция войск антироссийской коалиции выглядела так: 4 французские и 1 турецкая дивизия на правом фланге и 5 английских дивизий на левом фланге. Войска обеспечивались мощной огневой поддержкой корабельных орудий врага.

Вскоре перестрелка перешла в рукопашную схватку, в ходе которой на мосту через реку Альма конный разъезд под командованием ротмистра Уварова захватил курьера от французского маршала Сент-Арно. При первоначальном обыске в сумке пленного удалось обнаружить только письма частного характера. Однако наши солдаты проявили бдительность и доставили захваченные документы в армейскую канцелярию, где в присутствии князя Меншикова криптограф Степан Николаевич Мардарий дешифровал секретные донесения. Как выяснилось, французы применили стеганографический способ защиты информации – невидимые чернила[8]. С.Н. Мардарий высыпал на бумагу железные опилки и выставил с обратной стороны листа магнит. Железный порошок «расползся» по невидимым буквам. Русское командование узнало про обходной маневр дивизии генерала Боске и поспешило отвести войска с левого фланга. Битва была проиграна, но основные силы своей армии, благодаря успеху Мардария, А.С. Меншикову удалось сохранить. С.Н. Мардарий был пожалован «за усердие» орденом Станислава 4-й степени и чином титулярного советника. Это давало право на личное дворянство и соответствовало военному званию капитана. После окончания войны С.Н. Мардарий возвратился в Николаев. Он купил небольшой дом на углу Рождественской (Лягина) и Большой Морской, переселил в него пожилую мать, затем срочно отправился в Санкт-Петербург.

К тому времени в Российской империи начали активно действовать различные революционные организации, которые широко применяли шифры и стеганографию[9]. Правительство Российской империи резко нарастило штаты тайной полиции и привлекло к борьбе с революционерами криптографов. С.Н. Мардарий был назначен начальником специального департамента корпуса жандар-

мов. Офицеры относились к нему крайне уважительно, но за глаза все равно называли «чернильной душой». В задачу С.Н. Мардария и его коллег входило создание сети лабораторий по перлюстрации и графологической экспертизе всей подозрительной корреспонденции. В губернских почтовых отделениях и на таможнях появились чиновники, которые проверяли письма и всю печатную продукцию. Информация централизованно стекалась в Санкт-Петербург непосредственно к С.Н. Мардарию. Результаты работы перлюстраторов и криптографов не замедлили сказаться. «В период с 1864-го по 1880-й г. благодаря эффективной системе почтовой перлюстрации были арестованы 1038 боевиков-бомбистов, на таможнях изъято более 8000 наименований нелегальной литературы, 102 комплекта свинцового набора для подпольных типографий, 210 пудов пироксилина[10] и более 7000 единиц стрелкового оружия»[11].

В 1878 г. произошел резонансный случай, который сделал С.Н. Мардария практически легендарным человеком в полиции. Революционерка Вера Засулич, стрелявшая в санкт-петербургского градоначальника Ф.Ф. Трепова, находилась в тюрьме и ожидала решения суда присяжных. Она обратилась к тюремному начальству с просьбой отправить письмо родным, чтобы ей принесли теплую одежду. Разрешение было дано. Послание было передано для проверки в графологическую лабораторию. Статский советник С.Н. Мардарий находился в командировке, поэтому проверку осуществляли его подчиненные, которые ничего подозрительного в письме не обнаружили. Присяжные вынесли оправдательный приговор, и В. Засулич была отпущена из зала суда. Жандармы, которые хотели задержать революционерку по вновь открывшимся обстоятельствам преступления, не успели этого сделать. Извозчик, стоявший у парадного, быстро увез ее на конспиративную квартиру. Когда записка попала к Мардарию в руки, он быстро разобрался в ситуации. Послание на обратной стороне содержало информацию о плане побега, написанную водным раствором хлорида железа ($FeCl_3$) (В. Засулич принимала это вещество как лекарство). Такую записку можно прочесть, только обработав ее водным раствором тиоцианата калия, тогда все невидимые буквы становятся кроваво-красными.

С.Н. Мардарий внес вклад и в обучение российских криптографов. В 1893 г. Департамент полиции издал для внутреннего пользования написанный им учебник по графологии и криптографии. Появилась целая школа криминалистов-графологов и криптографов, которые вели активную работу по дешифрованию вражеских шифров во время Первой мировой войны. Криптографической де-

ятельности в этот период посвящена следующая статья. Умер Степан Николаевич Мардарий в августе 1917 г.[12]

Теперь вернемся к событиям Крымской войны. В отличие от нашей армии антироссийская коалиция на театре военных действий не располагала телеграфной связью. Как отмечает историк Ф. Риксон, «исход же многих сражений – на Инкерманских высотах, на Малаховом кургане и под Балаклавой – мог быть совсем иным, если бы донесения и приказы быстро передавались и умело зашифровывались»[13].

Одним из самых вопиющих случаев неправильной организации связи служит атака английской бригады легкой кавалерии в районе Балаклавы. 25 октября 1854 г. русские войска успешно атаковали английские позиции и захватили несколько английских артиллерийских батарей. Английское командование приняло решение немедленно отбить захваченные орудия. Операция была поручена бригаде легкой кавалерии (элитному подразделению английской армии, в котором служили многие аристократы) под командованием Джеймса Брюднелла, седьмого графа Кардигана. Атака была организована в спешке, атакующим не была точно поставлена боевая задача, а из-за отсутствия средств электросвязи передать уточняющие приказы, а тем более отменить атаку было невозможно. В результате бригада понесла большие потери. Этот эпизод Крымской войны был отражен в классическом стихотворении английского поэта Альфреда Теннисона[14].

Что касается криптоанализа, то у стран антироссийской коалиции с этим вопросом были большие проблемы. В середине XIX в. под давлением общественности были официально запрещены «черные кабинеты» в ведущих странах Европы. Бурные политические события середины XIX в. привели к ограничению абсолютной власти европейских монархов и власти их полицейских ведомств. Провозглашенные принципы свободы и равенства были несовместимы с цензурой переписки. Подглядывание и подслушивание якобы противоречили нравственным нормам поведения государств в отношениях друг с другом. И эти «кабинеты» везде были закрыты. В июне 1844 г. волна протестов со стороны общественности по поводу перлюстрации писем вынудила английское правительство прекратить перехват дипломатической переписки. В Австрии двери венского «черного кабинета» закрылись в 1848 г. А во Франции «черный кабинет», который уже со времен Великой французской революции дышал на ладан, в том же году также прекратил свое существование. Но это была только видимость. Очень скоро руководители государств поняли, что отказ от информационно-крипто-

графической поддержки наносил серьезный ущерб при принятии и реализации эффективных государственных решений. «Черные кабинеты» ушли в «подполье» и получили еще большее распространение. Они действовали неофициально, но очень скоро получили полную, хотя и секретную моральную и материальную поддержку со стороны государства[15].

Несмотря на закрытие «черного кабинета» в Великобритании, вся информация, связанная с шифрами и криптоанализом, жестко контролировалась властями. Так, например, английский криптограф-любитель Чарльз Бэббидж[16] в 1854 г. сумел разработать методику вскрытия одного из основных на тот момент шифров – шифра Виженера. Однако в связи с начавшейся Крымской войной английская секретная служба «потребовала от Бэббиджа, чтобы он сохранил свою работу в секрете»[17]. Таким образом, все лавры по поводу успешного криптоанализа шифра Виженера достались немцу Ф. Казиски, французам Э. Базери и маркизу де Виари, а также голландцу О. Кергхоффсу[18].

Подведем итоги. Крымская война стала первой «телеграфной» войной в истории. При этом российские шифры позволили сохранить секретность передаваемых сообщений от противника, а наши криптоаналитики достигли определенных успехов по вскрытию шифров антироссийской коалиции. При этом следует отметить, что телеграфная связь впервые стала использоваться в ходе боевых действий не только военными, но и журналистами. Сводки с театра военных действий оперативно доставлялись в столицы воюющих государств и публиковались в прессе. При этом журналисты, работавшие в войсках антироссийской коалиции, вынуждены были посылать свои сообщения в пункты, где были телеграфные аппараты (Турция и нейтральные страны), и только потом информация передавалась в европейские столицы. Наши журналисты были более оперативны, используя российские линии связи. Вот что об этом пишет историк Ф. Рикстон: «Телеграф стал играть ключевую роль в Крымской войне, правда в тактических действиях. Он дал возможность корреспондентам широко и подробно, как никогда прежде, освещать ход войны. Новости о потерях в результате боев, болезней и страшных морозов поступали в Париж, Лондон и Москву (тут Рикстон явно ошибается, столицей России тогда был Санкт-Петербург, и информация наших журналистов шла именно туда. – *Д. Л.*) с потрясающей воображение регулярностью. По мере затягивания войны растущее количество убитых, раненых и больных оказывало отрезвляющее воздействие на тех, кто поначалу ратовал за войну»[19].

Примечания

[1] Подробнее о ходе боевых действий см., в частности, сайты http://ru.wikipedia.org и http://kulichki.net/crimean.

[2] Подробнее об этом событии, биографии П.Л. Шиллинга, а также о влиянии появления и развития средств электросвязи на криптографию см.: *Бабаш А.В., Ларин Д.А.* История защиты информации в зарубежных странах: Учеб. пособие. М.: РИОР; ИНФРА-М, 2013; *Бабаш А.В., Баранова Е.К., Ларин Д.А.* Информационная безопасность. История защиты информации в России. М.: ИД КДУ, 2013; *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; *Ларин Д.А.* Революция в связи и криптографии. К 180-летию изобретения электрического телеграфа // Информационная безопасность банков. 2012. № 4 (7). С. 75–80.

[3] *Астрахан В.И., Гусев В.В., Павлов В.В., Чернявский Б.Г.* Становление и развитие правительственной связи в России. Орел: ВИПС, 1996. С. 29.

[4] Подробнее о российских шифрах XIX века см.: *Бабаш А.В., Баранова Е.К., Ларин Д.А.* Указ. соч.; *Соболева Т.А.* История шифровального дела в России. М.: ОЛМА-Пресс-Образование, 2002. 511 с.; *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.

[5] Подробнее о развитии телеграфной связи в Российской империи XIX в. см.: *Астрахан В.И., Гусев В.В., Павлов В.В., Чернявский Б.Г.* Указ. соч.; *Бабаш А.В., Баранова Е.К., Ларин Д.А.* Указ. соч.; *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.

[6] См.: *Соболева Т.А.* Указ. соч. С. 234.

[7] См.: Там же. С. 216.

[8] Как стало известно позже, невидимые чернила во время Крымской войны помимо французов активно применяли англичане, использовали подобный метод защиты информации и в русской армии.

[9] Подробнее о криптографическом противостоянии между революционерами и правоохранительными органами Российской империи см.: *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 20–70-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 деятельность организаций «Земля и Воля» и «Народная Воля» в России в 1876–1881 годах // Защита информации. INSIDE. 2005. № 6. С. 80–87; *Они же.* Криптографическая деятельность революционеров в России. 1881–1887 годы: агония «Народной Воли» // Защита

информации. INSIDE. 2006. № 2. С. 88–96; *Они же*. Криптографическая деятельность революционеров в России в 90-е годы XIX века: // Защита информации. INSIDE. 2006. № 4. С. 84–91; *Шанкин Г.П.* Криптографическая деятельность революционеров в России. Полиция против революционеров // Защита информации. INSIDE. 2008. № 2. С. 86–96; *Гольев Ю.И., Ларин Д.А., Шанкин Г.П.* Криптографическая деятельность революционеров в России на рубеже веков (1898–1900 годы) // Защита информации. INSIDE. 2008. № 4. С. 89–96; *Бабаш А.В., Баранова Е.К., Ларин Д.А.* Указ. соч. Гл. 5; *Соболева Т.А.* Указ. соч.; *Синельников А.В.* Шифры и революционеры России [Электронный ресурс] // Сайт «Хронос. Всемирная история в интернете». URL: http://www.hrono.ru/libris/lib_s/shifr00.html (дата обращения: 30.04.2014); Сайт «Математическая криптография» [Электронный ресурс]. URL: http://cryptography.ru/ (дата обращения: 30.04.2014).

[10] Взрывчатое вещество.

[11] См.: *Гаврилов С.* Душа чернильная [Электронный ресурс] // Николаевская областная интернет-газета «Новости N». URL: http://novosti-n.mk.ua/analitic/read/?id=765 (дата обращения: 30.04.2014).

[12] См.: Там же.
См.: *Гребенников В.В.* История криптологии и секретной связи [Электронный ресурс]. URL: http://www.cryptohistory.ru (дата обращения: 30.04.2014).

[13] *Риксон Ф.Б.* Коды, шифры, сигналы и тайная передача информации. М.: АСТ; Астрель; Владимир: ВКТ, 2011. С. 49.

[14] Более подробную информацию об этом сражении, а также текст стихотворения на английском языке и в русском переводе см.: Атака легкой бригады [Электронный ресурс] // Крымская война. URL: http://crimeawar.narod.ru/attack/tennyson.html (дата обращения: 08.01.2014), а также ряд других ресурсов Интернета.

[15] См.: *Бабаш А.В., Шанкин Г.П.* История криптографии. Ч. I. М.: Гелиос, 2002; *Кан Д.* Война кодов и шифров. М.: РИПОЛ КЛАССИК, 2004; *Kahn D.* The codebreakers. N. Y.: Macmillan Publ. Co., 1967.

[16] Ч. Бэббидж (1791–1871) занимался дешифрованием исторических шифров и был идеологом разработки первого в мире компьютера, который был создан уже после его смерти, подробнее см.: *Бабаш А.В., Гольев Ю.И., Ларин Д.А., Шанкин Г.П.* Криптографические идеи XIX века // Защита информации. Конфидент. 2004. № 1. С. 88–95; № 2. С. 92–96; *Сингх С.* Книга шифров: тайная история шифров и их расшифровки. М.: АСТ; Астрель, 2007.

[17] *Сингх С.* Указ. соч. С. 97.

[18] Подробнее о них см. статьи А.В. Бабаша и др.

[19] *Риксон Ф.Б.* Указ. соч. С. 48–49.

Междисциплинарные аспекты

А.А. Малюк, Н.Г. Милославская

НА ПУТИ К СОЗДАНИЮ ТЕОРИИ ЗАЩИТЫ ИНФОРМАЦИИ

В статье на опыте Российской Федерации рассматривается история развития исследований, направленных на формирование научно-методологических основ теории защиты информации. Приводятся полученные авторами результаты в области развития неформальной теории систем и подходов к созданию имитационных моделей процессов защиты информации в условиях неполноты и недостаточной достоверности исходных данных. На основе развитой структуры унифицированной концепции защиты информации последовательно рассмотрены проблемы, возникающие при практической реализации комплексной системы защиты.

Ключевые слова: теория защиты информации, концепция защиты информации, автоформализация знаний, модели защиты информации, оценка защищенности информации, требования к защите информации, системы защиты информации.

Интенсивное развитие и использование современных информационных технологий привели в настоящее время к серьезным качественным изменениям в экономической, социально-политической и духовной сферах общественной жизни. Человечество фактически переживает этап формирования нового информационного общества. Феномен резко возрастающего влияния информационно-коммуникационных технологий на формирование общества XXI в. был отмечен в Окинавской хартии глобального информационного общества, принятой лидерами «восьмерки» 22 июля 2000 г.

Отличительные черты информационного общества могут быть охарактеризованы следующим образом:

- существенный рост доли в валовом внутреннем продукте отраслей экономики, связанных с производством знаний, с созданием и внедрением наукоемких, в том числе информационных, технологий, других продуктов интеллектуальной деятельности, с оказанием услуг в области информатизации, образования, связи, а также поиска, передачи, получения и распространения информации;
- радикальное ускорение технического прогресса, превращение научных знаний в реальный фактор производства, повышения качества жизни человека и общества;
- участие значительной части трудоспособного населения в производственной деятельности, связанной с созданием и использованием информационных технологий, информации и знаний;
- глобализация экономической, политической и духовной сфер жизни общества.

В этих условиях на передний план экономического и социального развития выходят проблемы совершенствования систем информационного обеспечения всех сфер деятельности общества. Их решению в последние годы посвящаются интенсивные и крупномасштабные исследования и разработки.

Вместе с тем развитие информационного общества, помимо расширения созидательных возможностей, приводит и к росту угроз национальной и международной безопасности, связанных с нарушением установленных режимов использования информационных и коммуникационных систем, ущемлением конституционных прав и свобод граждан, распространением вредоносных программ, а также с использованием возможностей современных информационных технологий для осуществления враждебных, террористических и других преступных действий. В связи с этим особую остроту сегодня приобретает проблема обеспечения информационной безопасности, и прежде всего надежной защиты информации (предупреждения ее искажения или уничтожения, несанкционированной модификации, злоумышленного получения и использования).

Рассматриваемая проблема находится в центре внимания специалистов уже более 40 лет. Количество публикаций по различным аспектам обеспечения информационной безопасности и защиты информации исчисляется к настоящему времени несколькими тысячами, издаются специальные журналы, регулярно проводятся научные и практические конференции. Сегодня мы можем констатировать, что в процессе своего развития мировая «информационная» цивилизация пришла к формированию самостоятельного научно-технического направления «Информационная безопасность

и защита информации» и фактически к созданию новой важной сферы человеческой деятельности.

Становление научного направления «Информационная безопасность и защита информации» связано с именами таких отечественных ученых и специалистов, как В.А. Герасименко, П.Д. Зегжда, В.В. Кульба, О.Б. Макаревич, А.Г. Мамиконов, А.Г. Остапенко, С.П. Расторгуев, А.А. Стрельцов, А.А. Тарасов, Д.С. Черешкин, В.В. Шураков, А.Б. Шелков и др. Гуманитарные и правовые аспекты информационной безопасности наиболее выпукло представлены в трудах российских ученых: академика В.С. Степина, чл.-корр. РАН Ю.М. Батурина, В.И. Аршинова, И.Ю. Алексеевой, И.Л. Бачило, В.Н. Лопатина, М.А. Федотова и других.

И российские, и зарубежные специалисты единодушны в оценке чрезвычайной важности проблемы защиты. Естественно, что за истекшее после возникновения проблемы время коренным образом изменилось как представление о ее сущности, так и методологические подходы к решению. Указанные изменения происходили постепенно и непрерывно, поэтому всякая периодизация этого процесса в значительной мере носит искусственный характер. Тем не менее весь период активных работ по рассматриваемой проблеме довольно четко делится на три этапа.

Первые два этапа характеризуются экстенсивным подходом к решению задач защиты, которая в основном рассматривается в это время как защита от несанкционированного доступа к конфиденциальной информации. В течение этих этапов происходит постепенное расширение арсенала используемых средств защиты, причем как по их количеству, так и по разнообразию. При этом на втором этапе начинает получать распространение комплексное применение технических, программных и организационных средств. В целях регулирования правил защиты в ведущих странах начинают приниматься специальные законодательные акты.

Третий этап, характерный для настоящего времени, с полным правом может быть назван этапом комплексной защиты. Его особенность заключается в попытках обобщения всего имеющегося опыта теоретических исследований и практического решения задач защиты и формирования на этой основе научно-методологического базиса защиты информации. Иными словами, основная задача третьего этапа – перевод защиты информации на интенсивные способы, базирующиеся на строгой научной основе.

Кроме этого, в последнее время все более остро ставится проблема обеспечения информационной безопасности как органической совокупности решения задач защиты информации и защиты от информа-

ции. Не рассматривая здесь гуманитарные аспекты проблемы защиты от информации, а сосредоточившись только на обеспечении информационной безопасности автоматизированных систем, мы имеем достаточно веские основания утверждать, что решение проблемы защиты от информации (в частности от вредоносного программного обеспечения) может быть осуществлено на основе того же концептуально-методологического базиса, что и проблемы защиты информации. Это обстоятельство естественным образом подводит к заключению, что и проблему защиты от информации целесообразно включить в расширенное толкование понятия комплексной защиты информации.

Все это говорит о необходимости формирования научно-методологического базиса защиты как краеугольного камня интенсификации процессов обеспечения информационной безопасности.

Таким образом, на сегодняшний день можно выделить следующие наиболее острые проблемы, требующие своего решения:

- создание теоретических основ и формирование научно-методологического базиса защиты информации, позволяющих адекватно описывать процессы защиты и прогнозировать показатели защищенности в условиях значительной неопределенности и непредсказуемости проявления дестабилизирующих факторов;
- разработка научно обоснованных нормативно-методических документов по защите информации на базе исследования и классификации угроз информации и выработки стандартов требований к защите;
- стандартизация подходов к созданию систем защиты информации и рационализация схем и структур управления защитой на объектовом, региональном и государственном уровнях.

Исторический очерк (на опыте Российской Федерации)

Исторически можно выделить три этапа исследований, проводившихся в России и направленных на формирование теоретических основ защиты информации.

Первый этап (1991–2000 гг.) хронологически совпадает с широкомасштабным развертыванием в стране работ по защите информации и принятием Доктрины информационной безопасности Российской Федерации. Данный этап характеризуется эмпирическим подходом к решению практических задач обеспечения безопасности информации. Пользуясь терминологией, введенной известным российским ученым и общественным деятелем Н.Я. Данилевским

при рассмотрении им вопросов истории развития науки, назовем этот этап *«периодом собирания материалов»*. На данном этапе в целях формирования теоретико-методологической основы исследования осуществлялись изучение и анализ проводившихся отечественных и зарубежных работ, посвященных различным аспектам концептуально-методологических основ защиты информации и обеспечения информационной безопасности. В этот период А.А. Малюком совместно с профессором В.А. Герасименко был подготовлен и в 1997 г. издан первый российский учебник «Основы защиты информации»[1]. Учебник заложил основу для проведения исследований на следующем этапе, реализующем потребность привести накопленные сведения в определенную систему, учитывающую их сложную взаимосвязь.

Второй этап (2001–2008 гг.) был посвящен формированию на основе системного подхода концепции и содержания теории защиты информации в виде некоторого вербального описания процессов защиты. Таким образом, этот этап может быть назван концептуально-эмпирическим, или *«периодом искусственной системы»* (по Н.Я. Данилевскому). Этот этап естественным образом совпадает с разработкой и принятием Стратегии развития информационного общества в Российской Федерации. Обобщением исследований, проводившихся на этом этапе, явилось изданное в 2004 г. учебное пособие «Информационная безопасность: концептуальные и методологические основы защиты информации»[2].

Третий этап (2009–2013 гг.) охватывает исследования, логически завершающие формирование комплексного междисциплинарного подхода к обеспечению информационной безопасности и защиты информации. Данный этап может быть назван теоретико-концептуальным, или *«периодом естественной системы»*. Основные результаты исследований обобщены в изданной в 2012 г. монографии «Теория защиты информации»[3]. Необходимым компонентом, обеспечивающим комплексный взгляд на проблему, явились исследования, связанные с решением вопросов развития культуры информационной безопасности. Были исследованы вопросы этики в сфере информационных технологий, нашедшие отражение в монографии «Этика в сфере информационных технологий»[4], изданной в 2011 г.

В результате проводившихся исследований были сформированы:

1. *Концепция защиты информации*, построенная на идее перехода от экстенсивных к интенсивным методам решения проблем защиты и включающая структурированное описание среды защиты, всесторонний количественный анализ степени защищенности информации на объекте, научно обоснованное определение требуемо-

го уровня защиты на каждом конкретном объекте и в конкретных условиях его функционирования, построение оптимальных систем защиты на основе единой методологии. Главным здесь является использование упреждающей стратегии, опирающейся на научно обоснованное прогнозирование потенциально возможных ситуаций и вероятностей реализации широкого спектра угроз как случайного, так и преднамеренного характера. Переход от экстенсивных к интенсивным способам защиты информации ставит на повестку дня формирование нового научного направления – теории защиты информации. Фундаментальной основой теории защиты является научно-методологический базис, использующий неформально-эвристические методы. Исключительно важное значение для решения проблем защиты информации приобретают методы экспертных оценок, эвристического программирования, «мозгового штурма» и психоинтеллектуальной генерации, интегрированные в технологию автоформализации профессиональных знаний.

2. *Комплекс моделей и методологии адекватной оценки реальной защищенности информации, научного обоснования требований к защите, формирования оптимальной системы защиты и управления процессом ее функционирования*, основанные на системной классификации угроз, классификации множества вариантов потенциально возможных условий защиты информации, научно обоснованной типизации и стандартизации систем защиты, оптимизации управления их функционированием. Решение перечисленных проблем и выработка соответствующих требований осуществляется на основе структурно-логического анализа вариантов условий (ситуаций) защиты и структурированного их описания с широким применением методологии и методов неформальной теории систем, активно использующей процедуры экспертного оценивания. При этом степень адекватности моделей, получаемых экспертом, напрямую зависит от полноты и достоверности исходных данных, что требует разработки методологии оценки этой достоверности, а систематизация и обобщение подавляющего большинства данных – организации проведения массовой экспертизы.

3. *Концепция создания специализированных центров защиты информации*, направленная на решение проблемы совершенствования организационного обеспечения защиты информации в условиях ее интенсификации. Основными задачами данных центров являются:

- оказание услуг (аутсорсинг) по созданию и поддержанию функционирования систем защиты информации;
- обучение (подготовка, переподготовка, повышение квалификации) соответствующих кадров специалистов;

- сбор и формирование статистических данных об обеспечении информационной безопасности в пределах ареала конкретного центра, а также обмен этими данными с другими центрами.

4. *Концепция развития системы подготовки и переподготовки кадров в области обеспечения информационной безопасности*, решающая проблему совершенствования подготовки и расстановки соответствующих кадров специалистов. Проблема кадрового обеспечения в значительной степени определяет уровень информационной безопасности страны. В Российской Федерации к настоящему времени созданы основы системы подготовки и повышения квалификации специалистов в области информационной безопасности. Одно из основных мест в этой системе занимает подготовка кадров по направлению «Информационная безопасность». Дальнейшее развитие этой системы должно идти по пути формирования системы непрерывной подготовки и переподготовки кадров, совершенствования содержания и качества образования.

Ниже более подробно раскрывается содержание приведенных результатов. Основное внимание при этом обращается на следующие вопросы:

- современное состояние проблемы обеспечения информационной безопасности, место проблем защиты информации в общей совокупности информационных проблем современного общества;
- содержание теории защиты информации как междисциплинарного научного направления, являющегося теоретической основой интенсификации процессов защиты информации и обеспечения информационной безопасности;
- концепция защиты информации, основанная на идее комплексного подхода к реализации процесса защиты;
- методологические подходы к оценке защищенности информации, выработке требований к защите информации с учетом факторов, влияющих на уровень защиты, и потенциально возможных условий функционирования защищаемых систем и процессов, к построению систем защиты информации и управлению их функционированием;
- практические рекомендации по интенсификации процессов защиты информации, формированию современных организационных структур и системы кадрового сопровождения, обеспечивающих эффективную реализацию комплексного подхода к защите.

Современные проблемы защиты информации

Анализ достижений современной информатики и исторической ретроспективы развития подходов к информатизации ясно показы-

вает, что в настоящее время ведущие страны мира находятся в переходном периоде от индустриального этапа развития к информационному. На этом этапе главным стратегическим национальным ресурсом становятся информация и информационные технологии. Возрастание роли информации, информационных ресурсов и технологий в жизни граждан, общества и государства в XXI в. выводит вопросы информационной безопасности на первый план в системе обеспечения национальной безопасности, что требует разработки научно обоснованных подходов к их решению.

Объективные предпосылки для разработки общей теории безопасности (секьюритологии) пока не нашли своего удовлетворительного разрешения, хотя, по мнению ряда ученых (М.С. Алешенкова, Б.Н. Родионова, С.А. Филина, В.И. Ярочкина и других), системный подход к жизненно важной проблеме безопасности предопределяет необходимость формирования нового научного направления как систематизированного обобщенного знания обо всех аспектах безопасности, являющейся важнейшим условием выживания и развития человечества. Дискуссионный характер носят многие базовые понятия и определения, такие как «безопасность», «опасность», «угроза безопасности», «виды безопасности», «информационная безопасность» и др., что требует их всестороннего рассмотрения и уточнения с учетом современных научных воззрений. Приведенные в табл. 1 определения информационной безопасности, которые, хотя и могут быть приняты в плане практическом, не отражают специфики современного этапа формирования информационного общества.

Таблица 1

Актуальные определения информационной безопасности в соответствии с руководящими документами

Определение	**Источник**
1. Информационная безопасность – состояние защищенности информационной среды общества, обеспечивающее ее формирование, использование и развитие в интересах граждан, организаций, государства	Федеральный закон РФ от 4 июля 1996 г. № 85-ФЗ «Об участии в международном информационном обмене», ст. 2

Окончание таблицы 1

Определение	Источник
2. Информационная безопасность – защищенность ресурсов информационной системы от факторов, представляющих угрозу для: а) конфиденциальности; б) целостности; в) доступности	Code of practice for information security management. British Standard BS 7799, 1995; Information security management. Part 2 Specification for information security management systems. British Standard BS 7799, 1998 (Английский базовый стандарт «Практические правила управления информационной безопасностью»)
3. Информационная безопасность РФ – это состояние защищенности национальных интересов РФ в информационной сфере, определяющихся совокупностью сбалансированных интересов личности, общества, государства	Доктрина информационной безопасности РФ № Пр-1895 от 9 сентября 2000 г.
4. Под информационной безопасностью понимается защищенность информации и поддерживающей ее инфраструктуры от случайных или преднамеренных воздействий естественного или искусственного характера, чреватых нанесением ущерба владельцам или пользователям информации и поддерживающей инфраструктуры	По материалам Гостехкомиссии РФ[5]

Если рассматривать безопасность в качестве общенаучной категории, то представляется, что она может быть определена как некоторое качество той или иной системы, характеризующее, с одной стороны, ее способность противостоять дестабилизирующему воздействию внешних и внутренних угроз, а с другой – возможность возникновения угроз для элементов самой системы и внешней сре-

ды, связанных с ее функционированием. Интерпретация данной формулировки приводит к следующему определению информационной безопасности:

Информационная безопасность системы – это ее качество, характеризующее, с одной стороны, способность противостоять дестабилизирующему воздействию внешних и внутренних информационных угроз, а с другой – уровень угроз, которые создает ее функционирование для элементов самой системы и внешней среды.

Такое определение информационной безопасности отличается от распространенного определения, трактующего безопасность как состояние защищенности. Использование термина «состояние защищенности», на наш взгляд, не учитывает происходящих в последнее время изменений в подходах к созданию новых информационных технологий. При этом безопасность рассматривается не как некоторая надстройка, а как изначальный базис технологии, т. е. ее непременное качество. Таким образом, представление безопасности как качества более объективно характеризует способность системы противостоять тем или иным угрозам как внешнего, так и внутреннего характера. Учитывая приведенные соображения, предложенное определение можно считать достаточно полным и вполне корректным.

Однако для того чтобы служить более конкретным ориентиром в поиске решения проблем обеспечения информационной безопасности, оно нуждается в уточнении и детализации его основополагающих понятий. Из принятого нами определения естественным образом вытекает, что обеспечение информационной безопасности в общей постановке проблемы может быть достигнуто лишь при взаимоувязанном решении трех задач: защиты находящейся в системе информации от дестабилизирующего воздействия внешних и внутренних угроз; защиты элементов системы от дестабилизирующего воздействия внешних и внутренних информационных угроз; защиты внешней среды от информационных угроз со стороны рассматриваемой системы. Таким образом, обеспечение информационной безопасности может быть представлено двумя параллельными процессами: защитой информации и защитой от информации.

Как говорилось ранее, история развития подходов к решению проблем защиты информации может быть довольно четко разделена на три периода, которые могут быть названы соответственно эмпирическим, концептуально-эмпирическим и теоретико-концептуальным. Сущность подходов к защите в течение указанных периодов изменялась от выбора средств защиты на основе опыта через все более настоятельные попытки разработки и научного

обоснования методов оценки защищенности информации и синтеза оптимальных механизмов защиты к разработке в настоящее время основ теории защиты информации. Суть сегодняшнего этапа заключается в постановке задачи многоаспектной комплексной защиты и формировании унифицированной концепции защиты информации. Изменялись и применяемые средства защиты от функционально ориентированных механизмов до системы комплексной защиты и создания изначально защищенных информационных технологий.

Знаменательно, что предлагаемая периодизация истории развития подходов к защите информации в основном соответствует сформулированной еще в XIX в. российским ученым Н.Я. Данилевским системе ступеней или фазисов развития любой науки[6]. Н.Я. Данилевский выделял пять ступеней формирования области научного знания: собирания материалов, искусственной системы, естественной системы, частных эмпирических законов, общего рационального закона. Таким образом, эмпирический, концептуально-эмпирический и теоретико-концептуальный подходы соответствуют стадиям собирания материалов, искусственной и естественной систем. Можно предположить, что дальнейшее развитие теории защиты информации будет идти в направлении формулирования частных эмпирических законов и общего рационального закона, что соответствует созданию аксиоматической теории.

На сегодня же удалось разработать основы целостной теории защиты информации и этим самым подвести под реализацию защиты прочную научно-методологическую базу. Вместе с тем необходимо отметить, что до последнего времени системная реализация всех положений теории защиты сдерживается рядом серьезных трудностей, связанных с повышенным влиянием случайных факторов на процессы защиты информации, недостаточно четкой проработкой инструментальных средств решения задач анализа и синтеза систем и процессов защиты, с отсутствием значительной части исходных данных, необходимых для обеспечения решения названных задач.

Современный взгляд на защиту информации как на комплексную проблему неминуемо приводит к возрастанию значимости системных вопросов, связанных с процессом защиты (формирование общей политики защиты, оптимизация процессов проектирования и функционирования комплексных систем защиты, подбор, обучение и расстановка соответствующих кадров специалистов, сбор и аналитико-синтетическая обработка данных о функционировании реальных систем защиты информации). Таким образом, возникает

проблема системной увязки задач обеспечения информационной безопасности с остальными задачами решения информационных проблем общества. Проблема эта имеет как научно-методологические, так и организационные аспекты и в своем решении может базироваться на унифицированной концепции защиты информации, структура которой приведена на рис. 1. Идея данной концепции впервые была предложена В.А. Герасименко[7]. Концепция применима на всех трех уровнях защиты: компьютерном, объектовом, региональном (государственном). Она создает все необходимые объективные предпосылки для перехода к новому этапу в решении возникающих здесь задач – этапу интенсификации процессов защиты информации.

Переход от экстенсивных к интенсивным способам защиты информации означает целенаправленную реализацию всех достижений теории и практики защиты, которые в концентрированном виде отражены в унифицированной концепции, а именно: структурированное описание среды защиты, всесторонний количественный анализ степени защищенности информации на объекте, научно обоснованное определение требуемого уровня защиты на каждом конкретном объекте и в конкретных условиях его функционирования, построение оптимальных систем защиты на основе единой унифицированной методологии.

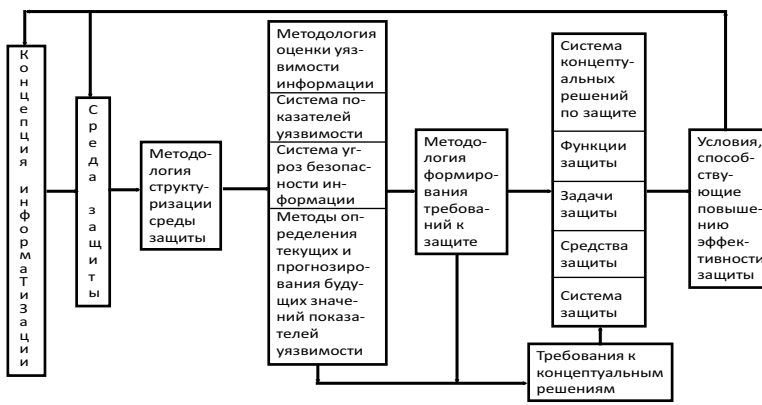


Рис. 1. Унифицированная концепция защиты информации

Научно-методологические основы интенсификации процессов защиты информации

Переход от экстенсивных к интенсивным способам защиты информации, составляющий, как это было показано выше, суть современного этапа формирования информационного общества, должен основываться на соответствующем научно-методологическом базисе. В связи с тем что процессы защиты информации подвержены сильному влиянию случайных факторов, методы классической теории систем оказываются практически непригодными для решения задач создания, организации и обеспечения функционирования систем защиты информации. Таким образом, возникает актуальная задача расширения арсенала классической теории за счет использования методов, позволяющих адекватно моделировать процессы, существенно зависящие от воздействия труднопредсказуемых факторов.

Наиболее подходящими для формирования методологического базиса теории защиты информации оказываются методы нечетких множеств, лингвистических переменных (нестрогой математики), неформального оценивания, неформального поиска оптимальных решений. При этом практически полное отсутствие на сегодняшний день систематизированных статистических данных функционирования реальных систем защиты информации выдвигает на передний план эвристическую составляющую этого базиса. Отсюда исключительное значение для решения проблем, поставленных в данной статье, приобретают методы экспертных оценок, эвристического программирования, «мозгового штурма» и психоинтеллектуальной генерации.

При использовании указанных подходов в задачах оценки состояния и прогнозирования уровня безопасности информации стратегия поиска решения, а также большинство этапов интерпретации результатов должны строиться в основном на неформальных знаниях эксперта и применяемых им интуитивных методах. В этих условиях единственным реальным способом создания моделей исследуемой ситуации на основе формализации алгоритмов аналитической деятельности может быть только автоформализация знаний эксперта, т. е. возникает проблема разработки технологии формализации экспертом своих профессиональных знаний.

Результатом автоформализации в этом случае являются новые сведения, которые эксперт получил в ходе эксперимента с моделями, и сами модели, отражающие его глубинные представления о структуре исследуемого процесса и присущих ему качественных и

количественных зависимостях. Последовательность и взаимосвязь этапов предлагаемого нами алгоритма автоформализации знаний показаны на рис. 2.

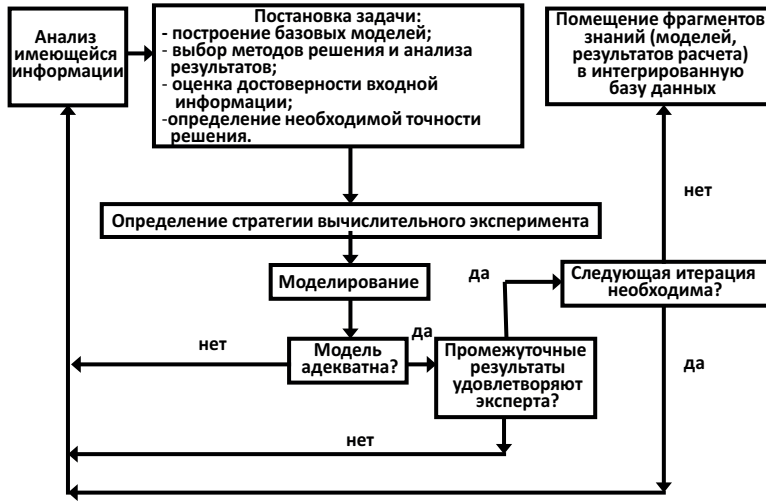


Рис. 2. Последовательность этапов автоформализации знаний

Исследование с его помощью проблемы защиты информации можно рассматривать как формальную систему, представляемую четырьмя показателями:

$$Z = < R_0, R_п, K, U >,$$

где R_0 – исходное состояние защищаемой системы, определяемое имеющимися в наличии данными;

$R_п$ – прогнозируемое состояние системы, соответствующее ее потенциальным возможностям противостоять угрозам безопасности информации;

K – знания о системе (элементарные и сложные модели, взаимосвязь между ними, ограничения на отдельные параметры и т. п.);

U – функция полезности системы, соразмеряющая эффективность функционирования и затраты на ее обеспечение.

Принципиальным элементом методологического базиса теории защиты информации является построение адекватных моделей изучаемых систем и процессов. Основой решения этой проблемы

может явиться обобщенная модель, блок-схема которой представлена на рис. 3. Модель оперирует со следующими параметрами: $\{K\}$ – множество показателей защищенности информации; $\{\Pi^{(с)}\}$ – множество параметров внешней среды, оказывающих влияние на функционирование автоматизированной системы; $\{R^{(с)}\}$ – множество ресурсов системы, участвующих в обработке защищаемой информации; $\{\Pi^{(у)}\}$ – множество внутренних параметров системы, которыми можно управлять непосредственно в процессе обработки защищаемой информации; $\{\Pi^{(в)}\}$ – множество внутренних параметров системы, не поддающихся непосредственному управлению, но поддающихся воздействию (например, в процессе реорганизации или совершенствования компонентов системы); $\{S^{(у)}\}$ и $\{R^{(у)}\}$ – множества средств и ресурсов текущего управления; $\{S^{(в)}\}$ и $\{R^{(в)}\}$ – множества средств и ресурсов воздействия; $\{R^{(о)}\}$ – множество общих ресурсов управления.

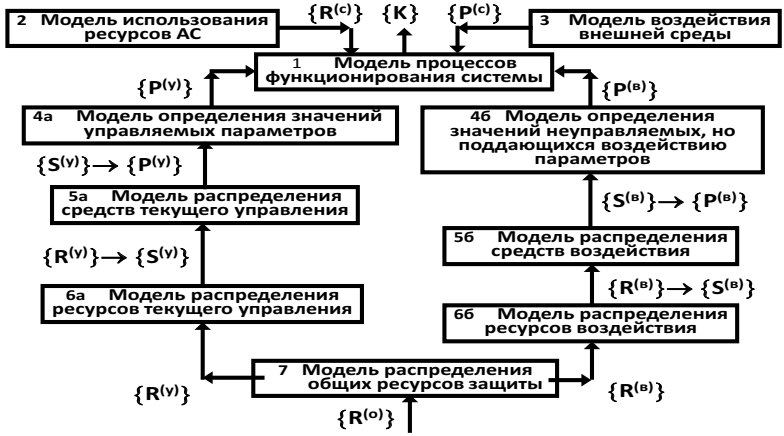


Рис. 3. Обобщенная модель процессов защиты информации

Для решения с помощью этой модели задач анализа, т. е. для определения значений показателей защищенности информации, можно использовать следующее обобщенное выражение:

$$\{K\} = F[\{\Pi^{(у)}\}, \{\Pi^{(в)}\}, \{R^{(с)}\}, \{\Pi^{(с)}\}],$$

Задачи синтеза в общем виде могут быть представлены следующим образом:

найти такие $\{R^{(у)}\}$ и $\{R^{(в)}\}$ ($\{R^{(у)}\} + \{R^{(в)}\} \le \{\overline{R}^{(0)}\}$ ($\{\overline{R}^{(0)}\}$ – заданные ресурсы), чтобы при заданных $\{R^{(с)}\}$ и $\{\Pi^{(с)}\}$ выполнялось условие $\{K\} \to \max$;

выбрать такие $\{R^{(у)}\}$ и $\{R^{(в)}\}$, чтобы при заданных $\{R^{(с)}\}$ и $\{\Pi^{(с)}\}$ условие $\{K\} \ge \{\overline{K}\}$ ($\{\overline{K}\}$ – заданный уровень защищенности) выполнялось при $\{R^{(о)}\} = \{R^{(у)}\} + \{R^{(в)}\} \to \min$.

Таким образом, задачи управления сводятся к оптимизации распределения $\{R^{(у)}\}$, $\{S^{(у)}\}$, $\{R^{(в)}\}$, $\{S^{(в)}\}$.

Нетрудно видеть, что возможны следующие модификации обобщенной модели:

модель функционирования системы при отсутствии управления защитой информации (такая модель позволяет лишь определять значения показателей защищенности информации, т. е. решать задачи анализа);

модель текущего управления защитой информации, основу которого составляет оптимизация использования средств защиты, непосредственно включенных в состав системы;

модель управления ресурсами, выделенными на защиту информации, которая дополнительно к предыдущим задачам позволяет оптимизировать процесс формирования средств текущего управления защитой информации;

модель управления средствами воздействия на параметры, не допускающие текущего управления, но поддающиеся воздействию;

модель управления ресурсами, выделенными на развитие системы;

полная модель защиты, которая дополнительно ко всем возможностям, рассмотренным выше, позволяет оптимизировать использование всех ресурсов, выделенных на защиту информации.

Эффективность практического использования данной модели существенно зависит от представительности и адекватности массивов статистических данных, позволяющих определять функциональные зависимости, устанавливающие взаимосвязи показателей защищенности информации, параметров систем защиты и размеров ресурсов, вкладываемых в реализацию процессов защиты. В связи с этим рассматриваемая модель может применяться только в совокупности с неформальными методами анализа и прогнозирования, в частности, с использованием рассмотренного выше алгоритма автоформализации знаний эксперта-аналитика. Отсюда понятно также, что принципиальное значение в этих условиях имеет решение проблемы организационного обеспечения всего комплекса работ по защите информации, позволяющего реализовать сбор, накопление и систематизацию исходных данных.

Основными результатами развития теории защиты информации являются также введение понятия «стратегия защиты» и создание единого инструментально-методологического базиса ее реализации. Проведенные исследования показывают, что с учетом требуемого уровня защиты и степени свободы действий при ее организации целесообразно выделить три базовые стратегии защиты информации: оборонительную, наступательную и упреждающую. При этом каждая из этих стратегий может быть эффективно реализована в рамках унифицированной концепции защиты информации (рис. 1).

Методология оценки защищенности информации

В настоящее время известно большое количество разноплановых угроз различного происхождения, таящих в себе различную опасность для информации. На протяжении нескольких десятков лет не прекращаются попытки формирования, исследования и классификации возможно более полного множества угроз. Актуальность решения этой проблемы заключается в необходимости строгого определения значений показателей защищенности информации, что должно позволить построить адекватные модели процессов и механизмов ее защиты.

В целях обоснования структуры и содержания системы показателей защищенности информации, исследования влияния на них различных параметров угроз, разработки комплекса моделей и методологии оценки на их основе защищенности информации проведем системную классификацию угроз (табл. 2). Следует иметь в виду, что представленные в табл. 2 параметры классификации находятся в сложных взаимосвязях.

Для системной оценки защищенности информации можно использовать систему показателей, структурированную по видам защиты информации и видам дестабилизирующего воздействия на нее. Основными параметрами, определяющими вероятность нарушения защищенности информации, примем: количество и типы структурных компонентов системы, количество и типы случайных дестабилизирующих факторов, количество и типы злоумышленных дестабилизирующих факторов, число и категории нарушителей, виды защищаемой информации.

Таблица 2

Системная классификация угроз информации

Параметры классификации	**Содержание параметров**	**Деструктивные действия**
1. Виды угроз	1.1. Нарушение физической целостности	Уничтожение (искажение)
	1.2. Нарушение логической структуры	Искажение структуры
	1.3. Нарушение содержания	Несанкционированная модификация
	1.4. Нарушение конфиденциальности	Несанкционированное получение
	1.5. Нарушение права собственности	Присвоение чужого права
2. Природа происхождения угроз	2.1. Случайная	Отказы Сбои Ошибки Стихийные бедствия Побочные влияния
	2.2. Преднамеренная	Злоумышленные действия людей
3. Предпосылки появления угроз	3.1. Объективные	Количественная недостаточность элементов системы Качественная недостаточность элементов системы Разведорганы иностранных государств
	3.2. Субъективные	Промышленный шпионаж Уголовные элементы Недобросовестные сотрудники

Окончание таблицы 2

Параметры классификации	Содержание параметров	Деструктивные действия
4. Источники угроз	4.1. Люди	Посторонние лица Пользователи Персонал
	4.2. Технические устройства	Регистрации Передачи Хранения Переработки Выдачи
	4.3. Модели, алгоритмы, программы	Общего назначения Прикладные Вспомогательные
	4.4. Технологические схемы обработки	Ручные Интерактивные Внутримашинные Сетевые
	4.5. Внешняя среда	Состояние атмосферы Побочные шумы Побочные сигналы

Наибольшую угрозу в современных условиях представляют злоумышленные действия людей. Защищенность информации для таких действий может быть оценена на основе представления объекта защиты в виде совокупности нескольких рубежей обороны следующим выражением:

$$P_{ijkl} = P^{(д)}_{ikl} \cdot P^{(к)}_{ijl} \cdot P^{(н)}_{ijkl} \cdot P^{(и)}_{ijl},$$

где P_{ijkl} – вероятность несанкционированного получения информации нарушителем k-й категории по j-му каналу несанкционированного получения информации в l-й зоне i-го структурного компонента системы;

$P^{(д)}_{ikl}$ – вероятность доступа нарушителя k-й категории в l-ю зону i-го компонента;

$P^{(к)}_{ijl}$ – вероятность наличия (проявления) j-го канала в l-й зоне i-го компонента;

$P^{(н)}_{ijkl}$ – вероятность доступа нарушителя k-й категории к j-му каналу в l-й зоне i-го компонента при условии доступа нарушителя в зону;

$P^{(и)}_{ijl}$ – вероятность наличия защищаемой информации в j-м канале в l-й зоне i-го компонента в момент доступа туда нарушителя.

Аналогичным образом могут быть определены и другие показатели защищенности, связанные с нарушением целостности и доступности информации. При этом множество всех разновидностей различных показателей определяется декартовым произведением чисел, характеризующих количество значений всех значащих параметров. Из этого множества особо могут быть выделены два показателя: первый (базовый), характеризующий защищенность информации в одном структурном компоненте системы при однократном проявлении одного дестабилизирующего фактора и относительно одного потенциального нарушителя, второй (общий), характеризующий защищенность информации в целом по всем потенциально возможным дестабилизирующим факторам относительно всех потенциально возможных нарушителей. Все другие возможные показатели будут являться частично обобщенными.

Базовый показатель защищенности от злоумышленных действий, имеющих место в пяти различных зонах объекта информатизации (внешней неконтролируемой зоне, зоне контролируемой территории, зоне помещений системы, зоне ресурсов системы и зоне баз данных), может быть определен по следующему выражению:

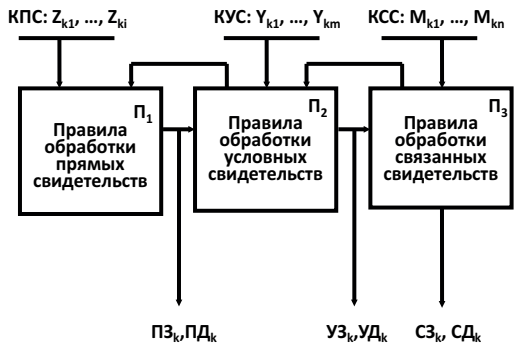
$$P^{(б)}_{ijk} = 1 - \prod_{i=1}^{5} (1 - P_{ijkl}) = 1 - \prod_{i=1}^{5} \left[1 - P^{(д)}_{ikl} \cdot P^{(к)}_{ijl} \cdot P^{(н)}_{ijkl} \cdot P^{(и)}_{ijl} \right]$$

Следует отметить, что во всех этих выражениях, структурирующих оценку защищенности информации, присутствуют показатели, представляющие собой вероятности реализации тех или иных событий. Значения этих показателей при отсутствии достаточного статистического материала могут быть получены только экспертным путем с использованием описанной выше технологии автоформализации знаний. При этом исключительное значение приобретает оценка достоверности данных, опираясь на которые эксперт-аналитик принимает то или иное решение.

Определим достоверность как «уровень разумной уверенности в истинности некоего высказывания, который удовлетворяет некоторым правилам непротиворечивости и в соответствии с этими правилами формально может быть выражен числом»[8]. Используя идею байесовского подхода, можно поставить вопрос о достоверности фрагментов интегрированной базы данных (базы данных,

базы знаний и базы моделей) эксперта-аналитика, рассматривая любой фрагмент как гипотезу, а фрагменты, с которыми он связан, как свидетельства относительно фрагмента-гипотезы. Тогда каждый вновь поступающий в интегрированную базу данных фрагмент (НФЗ – новый фрагмент знаний) может быть представлен как пара НФЗ = <З,Д>, где З – значение фрагмента, а Д – его достоверность.

НФЗ, будучи включенным в интегрированную базу данных, взаимодействует с уже содержащимися в ней фрагментами и гипотезами, изменяя как их значения, так и достоверности. Эта реакция достаточно сложна и вызывает модификацию значений и достоверностей всех старых фрагментов, так или иначе связанных с вновь поступившим. Для описания процесса модификации введем понятия «системное значение» и «системная достоверность фрагмента», определяемые с учетом всех свидетельств, содержащихся в интегрированной базе данных. Модификация значения и достоверности фрагмента при изменении состава свидетельств может быть осуществлена с использованием алгоритма, блок-схема которого изображена на рис. 4.



КПС – кортеж прямых свидетельств для данного фрагмента
КУС – кортеж условных свидетельств
КСС – кортеж связанных свидетельств
ПЗ, ПД – значение и достоверность фрагмента с учетом всех ПС
УЗ, УД – значение и достоверность фрагмента с учетом всех УС
СЗ, СД – системные значения и достоверности фрагмента

Рис. 4. Блок-схема алгоритма вычисления системного значения и системной достоверности фрагмента интегрированной базы данных

Для обработки свидетельств в данном алгоритме модификации могут быть применены различные методы. Наиболее удоб-

ным представляется подход, основанный на применении методов фильтрации. При этом общая постановка задачи – это система двух уравнений, первое из которых описывает структуру и динамику исследуемого процесса защиты, а второе – определяет механизм образования данных, доступных для эксперта-аналитика:

$$x(k+1) = F[x(k), w(k), d(k)],$$

$$y(k) = H[x(k), v(k), d(k)],$$

где $x(k)$ – вектор состояния исследуемого процесса;

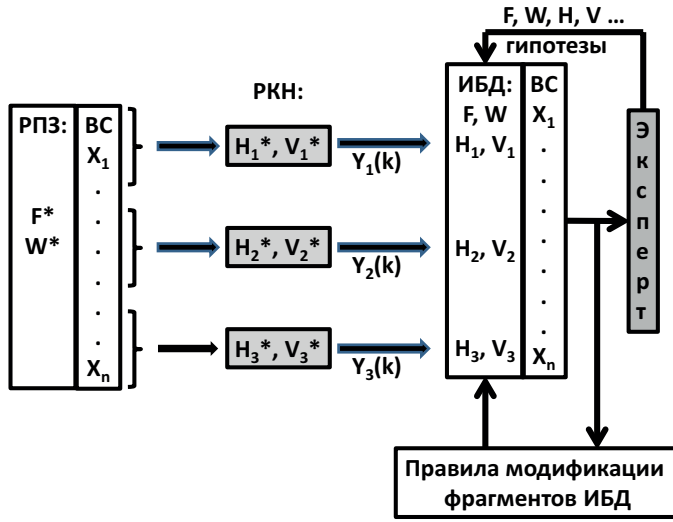
$w(k)$ – случайный вектор шумов исследуемого процесса, связанных с погрешностями методов моделирования;

$y(k)$ – вектор наблюдения;

$v(k)$ – случайный вектор шумов наблюдения, связанных с погрешностями канала получения информации (w и v не коррелированы);

$d(k)$ – вектор вариативности, характеризующий текущее состояние и структуру процесса и канала получения информации (при этом отказ рассматривается как изменение параметров или структуры).

Модель информационной среды эксперта для данного подхода показана на рис. 5.



РПЗ – реальный процесс защиты
ВС – вектор состояния
РКН – реальный канал наблюдения
ИБД – интегрированная база данных

Рис. 5. Модель информационной среды эксперта

Алгоритм, представленный на этом рисунке, позволяет эксперту каждый раз при поступлении новых данных y(k) рекуррентно модифицировать оценку значения вектора состояния $x(k)$ и корреляционную матрицу ошибок $P(k)$, характеризующую достоверность этой оценки с учетом всех поступивших на данный момент наблюдений $y(k)$, а также динамики и структуры процесса и каналов получения информации.

Методология определения требований к защите информации

Переход от экстенсивных к интенсивным способам защиты информации требует строго научного обоснования конкретных требований к защите. Данные требования определяются характером, видом и объемом обрабатываемой информации, продолжительностью ее пребывания в системе, технологией обработки и организацией информационно-вычислительного процесса, структурой и этапом жизненного цикла защищаемой системы.

В современных условиях наиболее рациональным подходом к выработке этих требований представляется подход, основанный на выделении некоторого количества типовых систем защиты и разработке рекомендаций по их использованию. На сегодняшний день в целях типизации систем защиты информации разработано множество регламентирующих документов, определяющих критерии оценки защищенности автоматизированных систем и механизмы защиты, которые должны использоваться при обработке информации различной степени конфиденциальности.

Наличие данных документов создает достаточную базу для организации защиты информации на регулярной основе. Однако с точки зрения современной постановки задачи защиты (интенсификация процессов и комплексный подход) они имеют ряд принципиальных недостатков, так как ориентированы на защиту информации только в средствах вычислительной техники и учитывают далеко не все факторы, оказывающие существенное влияние на защищенность информации. Кроме того, их научное обоснование оставляет желать лучшего.

В целях совершенствования методик определения требований к защите информации и типизации на этой основе систем защиты необходимо решить следующую последовательность задач:

разработка методов оценки параметров защищаемой информации;

формирование перечня и классификация факторов, влияющих на требуемый уровень защиты;

структуризация возможных значений факторов;

структуризация поля потенциально возможных вариантов условий защиты;

оптимальное деление поля возможных вариантов на типовые классы;

структурированное описание требований к защите в пределах выделенных классов.

Для оценки параметров защищаемой информации можно использовать показатели, характеризующие ее как обеспечивающий ресурс для решения прикладных задач и как объект труда для процесса информационного обеспечения решаемых задач. К показателям первого вида целесообразно отнести важность, полноту, адекватность, релевантность и толерантность информации. В качестве основных характеристик второго вида можно использовать способ кодирования и объем информации.

Оценка важности, полноты и адекватности информации базируется на неформально-эвристических методах и использовании лингвистических переменных. В результате нами был разработан ряд классификаций указанных параметров, позволяющих практически решать задачи оценки защищаемой информации.

Важность информации предложено оценивать по двум группам критериев: по назначению и по условиям ее обработки. Полноту информации целесообразно оценивать на основе формирования объектно-характеристических таблиц (информационного кадастра объекта). Адекватность информации определяется объективностью ее генерирования и продолжительностью интервала времени между моментом генерирования и моментом оценивания адекватности.

В целях формирования возможно более полного множества факторов, влияющих на требуемый уровень защиты, и возможно более адекватного определения степени этого влияния разработан подход, базирующийся на неформально-эвристических методах с широким привлечением знаний, опыта и интуиции компетентных и заинтересованных специалистов.

Практическое использование этого подхода позволило выделить в общей сложности 17 факторов, каждый из которых может принимать одно из четырех значений. При этом общее число возможных вариантов условий защиты превышает $1{,}7 \cdot 10^{10}$, что, естественно, делает задачу определения конкретных требований к защите информации практически неразрешимой. Данная ситуация приводит к необходимости деления всего множества возможных вариантов на некоторое число классов, которое можно было бы ис-

пользовать в дальнейшем для практического решения поставленной задачи.

Осуществление такой классификации фактически является задачей формирования необходимого и достаточного набора типовых систем защиты информации. На основании теоретического, эмпирического и теоретико-эмпирического подходов к решению этой проблемы предлагается классификационная структура типовых систем защиты информации, содержащая пять основных и пять дополнительных классов (табл. 3).

Таблица 3

Классификационная структура типовых систем защиты информации

Уровень защиты	**Стратегия защиты**		
	оборонительная	наступательная	упреждающая
Слабый	1	-	-
Средний	2	2н	-
Сильный	3о	3	3у
Очень сильный	-	4	4у
Особый	-	5н	5

Общее число вариантов условий при такой классификации в случае надлежащего построения вычислительного алгоритма оказывается вполне подъемным для современной компьютерной техники.

Методология формирования систем защиты информации

Из предшествующего рассмотрения ясно, что все ресурсы, выделяемые в системе для защиты информации, должны быть объединены в единую, функционально самостоятельную подсистему. С точки зрения организационного построения такая подсистема (назовем ее системой защиты информации – СЗИ) представляет собой совокупность механизмов обеспечения защиты информации, механизмов управления механизмами обеспечения защиты и механизмов общей организации работы системы. Общая структур-

ная схема СЗИ при таком подходе к ее организации представлена на рис. 6.

Рис. 6. Общая структурная схема системы защиты информации

Важнейшее значение для обеспечения надежности и экономичности защиты информации имеют типизация и стандартизация СЗИ. Анализ концепции защиты информации и возможных подходов к архитектурному построению СЗИ показывает, что целесообразно выделить три уровня типизации и стандартизации: высший – уровень СЗИ в целом, средний – уровень компонентов СЗИ и низший – уровень проектных решений по средствам и механизмам защиты.

С целью создания объективных предпосылок для типизации и стандартизации на высшем и среднем уровнях на основании эмпирического подхода может быть предложена системная классификация СЗИ. В соответствии с этой классификацией все СЗИ в зависимости от уровня обеспечиваемой защиты могут быть разделены на четыре категории: системы слабой защиты, системы сильной защиты, системы очень сильной защиты, системы особой защиты, а в зависимости от активности реагирования на несанкционированные действия – на три типа: пассивные СЗИ, полуактивные СЗИ и активные СЗИ. Для формирования полного множества необходимых СЗИ должен быть принят во внимание также и тип системы, для которой эта СЗИ предназначена. В соответствии с современными представлениями можно различать следующие системы: персональный компьютер, используемый локально (ПК); групповой компьютер, используемый локально (ГК); вычислительный центр предприятия, учреждения, организации (ВЦП); вычислительный центр коллективного пользования (ВЦКП); локальная вычислительная сеть (ЛВС); «слабо» распределенные (в пределах населенного пункта, небольшой территории) вычислительные сети (СВС);

региональные вычислительные сети (РВС); глобальные вычислительные сети (ГВС). С учетом этого итоговая классификация СЗИ будет иметь вид, приведенный в табл. 4.

Таблица 4

Итоговая классификация СЗИ

Вариант СЗИ / Тип АС	1 Слабой защиты Пассивные	2 Сильной защиты Полуактивные	2а Сильной защиты Активные	3 Очень сильной защиты Активные	3а Очень сильной защиты Полуактивные	4 Особой защиты Активные
ПК	Ц (1)	Д/Ц* ($1_а$)	-	-	-	-
ГК	Ц* ($2_а$)	Ц (2)	Д/Ц* ($2_б$)	-	-	-
ВЦП	Д* ($3_а$)	Ц (3)	Д/Ц* ($3_б$)	Д* ($3_в$)	-	-
ВЦКП	-	-	Ц (4)	Ц* ($4_а$)	Ц* ($4_б$)	Д* ($4_в$)
ЛВС	-	Ц* ($5_а$)	Ц (5)	Д* ($5_б$)	-	-
СВС	-	Ц* ($6_а$)	Ц* ($6_б$)	Ц (6)	Ц* ($6_в$)	Д* ($6_г$)
РВС	-	-	Ц* ($7_а$)	Ц (7)	Д* ($7_б$)	Ц* ($7_в$)
ГВС	-	-	-	Ц (8)	-	Ц* ($8_а$)

Ц – целесообразно
Д – допустимо
* – в отдельных случаях

Типизация и стандартизация на среднем уровне предполагает разработку типовых проектов структурно и функционально ориентированных компонентов СЗИ. В качестве наиболее перспективного варианта покомпонентной типизации и стандартизации СЗИ предлагается использовать подход, основанный на так называемой семирубежной модели.

Типизация и стандартизация на низшем уровне предполагает разработку типовых проектных решений по практической реализации средств защиты информации (технических, программных, организационных).

Проектирование СЗИ заключается в создании рациональных механизмов обеспечения защиты информации и механизмов

управления ими. В зависимости от уровня конфиденциальности защищаемой информации, условий, в которых создается система защиты, а также с учетом предложенной выше классификации СЗИ структура возможных подходов к проектированию системы будет включать шесть различных вариантов, от использования типовых СЗИ до разработки индивидуального проекта системы с применением индивидуальных средств защиты.

При решении задач анализа и оценки СЗИ для целей адаптации к различным условиям защиты и управления их развитием могут применяться рассмотренные выше методы моделирования процессов защиты информации. При этом необходимость учета множества факторов, влияющих на защиту и находящихся в сложном динамичном взаимодействии, приводит к представлению системы защиты как многокритериального развивающегося объекта, характеризуемого критерием эффективности Э(X):

$$Э(x_1, ..., x_m) = \sum_{i=1}^{m} \alpha\,(x_1, ..., x_m)\, R(x_i)\,,$$

где $Э(x_1, ..., x_m)$ – монотонно возрастающий критерий, заданный на множестве значений x_i (i=1,..., m), $0 \le x_i \le 1$;

$\alpha\,(x_1, ..., x_m)$ – непрерывные положительные функции, называемые коэффициентами весомости показателей;

$$\sum_{i=1}^{m} \alpha\,(x_1, ..., x_m) = 1;$$

$R(x_i)$ – ранговая функция полезности (РФП), определяемая как значение критерия на диагонали, т. е. $R(t) = Э(x_1 = t, ..., x_m = t)$.

Определив еще функцию полезности при анализе рисков (ФПР) как $U(t) = L(x_1 = t, ..., x_m = t)$, где $L(x_1, ..., x_m)$ – каноническое описание поверхности уровня, определяемое условием Э(X) = const, можно, используя РФП и ФПР, которые получаются на основании фактических данных о группировании реализации СЗИ по классам в соответствии с тем, как было изложено выше, ставить задачу синтеза критерия Э(X).

Отметим еще в завершение этого раздела некоторые проблемы управления процессами функционирования систем защиты. Здесь целесообразно выделить основные макропроцессы управления, к которым могут быть отнесены планирование, оперативно-диспетчерское управление, календарно-плановое руководство и обеспечение повседневной деятельности. Схемы практической реализации

всех этих процессов должны учитывать особенности краткосрочного, среднесрочного и долгосрочного управления.

Особое значение в процессах управления имеет регулярно осуществляемый контроль защищенности, который складывается из собственно контроля защищенности информации и контроля функционирования механизмов защиты. Технология контроля защищенности должна предусматривать контроль состояния параметров, определяющих значения контролируемых показателей, контроль проявления типовых нарушений правил защиты информации, а также комбинированный контроль по параметрам и типовым нарушениям.

Совершенствование организационных подходов и кадрового обеспечения решения перспективных проблем защиты информации

Анализ обобщенных итогов развития теории и практики защиты информации приводит к выводу о том, что наиболее вероятными перспективами их дальнейшего развития являются:

совершенствование теоретических основ защиты;

практическая реализация идеи интенсификации защиты информации и перевод ее на индустриальную основу;

постепенная трансформация задачи защиты информации (в основном обеспечения так называемой компьютерной безопасности, или кибербезопасности) в задачу обеспечения информационной безопасности объектов, регионов и государства в целом.

Центральное место в ближайшей перспективе займут проблемы перехода от экстенсивных к интенсивным методам реализации процессов защиты информации. Особую роль в совершенствовании организационного обеспечения защиты информации в условиях ее интенсификации должен играть аутсорсинг на основе создания специализированных центров защиты. Основными задачами данных центров должны стать:

оказание услуг предприятиям, учреждениям, иным организациям по созданию и поддержанию функционирования в них систем защиты информации;

обучение (подготовка, переподготовка, повышение квалификации) соответствующих кадров специалистов;

сбор и формирование статистических данных об обеспечении информационной безопасности в пределах ареала конкретного центра, а также обмен этими данными с другими центрами.

Следует отметить, что в России концепция центров защиты информации на сегодняшний день практически реализована в виде сети региональных учебно-научных центров по проблемам информационной безопасности в системе высшей школы.

Для практического применения рассмотренных в статье основ теории защиты информации принципиальное значение имеет формирование репрезентативных баз, исходных данных анализа и прогнозирования ситуаций защиты. Эта деятельность может рассматриваться как одна из важнейших макрозадач центров защиты информации. При этом так как подавляющее большинство исходных данных носит настолько неопределенный характер, что может быть получено (по крайней мере, в настоящее время и в обозримом будущем) только неформально-эвристическими методами, возникает задача организации и осуществления процедуры непрерывной массовой экспертизы.

Решение современных проблем защиты информации и более общих проблем обеспечения информационной безопасности в определяющей степени зависит также от подготовки и расстановки соответствующих кадров специалистов. В Российской Федерации к настоящему времени развернута система подготовки и повышения квалификации специалистов в области информационной безопасности. В общей сложности подготовку специалистов по тем или иным аспектам безопасности информации ведут сейчас уже более 100 вузов. Большое значение для совершенствования подготовки специалистов имеет развитие международного сотрудничества в этой области. В связи с этим нельзя не отметить положительного опыта регулярного проведения Международной конференции по образованию в области информационной безопасности в рамках деятельности международной федерации по обработке информации. Странами-организаторами этой конференции в разное время выступали Швеция, Австралия, США, Россия, Бразилия, Швейцария, Новая Зеландия.

Заключение

Рассматривая проблему интенсификации подходов к обеспечению безопасного развития информационного общества, мы можем констатировать, что на сегодня достигнуты следующие результаты.

Обосновано и в содержательном плане сформировано новое научное направление – теория защиты информации, базирующееся на структурированном описании среды защиты, всестороннем

количественном анализе степени защищенности информации на объекте, научно обоснованном определении требуемого уровня защиты на конкретных объектах в любых возможных условиях их функционирования, построении оптимальных систем защиты на основе единой унифицированной методологии.

Сформирован научно-методологический базис теории защиты информации в виде неформальной теории систем, использующий достижения методов нечетких множеств, лингвистических переменных, экспертных оценок, неформального оценивания, неформального поиска оптимальных решений.

Предложены методологические подходы к разработке политики безопасности в условиях неполноты и недостоверности исходных статистических данных, структурирован процесс создания оптимальных систем защиты информации в виде кортежа концептуальных решений, составляющих существо унифицированной концепции защиты информации.

Разработаны системные классификации угроз безопасности информации, потенциально возможных условий защиты, основных типов архитектурного построения систем защиты, являющиеся основой масштабной стандартизации систем и процессов защиты информации.

Предложен новый подход к формированию организационного обеспечения решения проблем информационной безопасности на основе аутсорсинга в области защиты информации и создания для этой цели специализированных региональных центров.

Дальнейшие исследования предполагается проводить в направлении формирования аксиоматической теории защиты информации и разработки поведенческих моделей нарушителей, позволяющих повысить эффективность прогнозирования ситуаций защиты и дестабилизирующих факторов.

Примечания

[1] См.: *Герасименко В.А., Малюк А.А.* Основы защиты информации: Учебник. М.: МИФИ, 1997.

[2] См.: *Малюк А.А.* Информационная безопасность: концептуальные и методологические основы защиты информации: Учеб. пособие. М.: Горячая линия Телеком, 2004.

[3] См.: *Малюк А.А.* Теория защиты информации. М.: Горячая линия Телеком, 2012.

[4] См.: *Малюк А.А., Полянская О.Ю., Алексеева И.Ю.* Этика в сфере информационных технологий. М.: Горячая линия Телеком, 2011.

[5] См.: *Байбурин В.Б.* Введение в защиту информации: Учеб. пособие / В.Б. Байбурин, М.Б. Бровкова и др. М.: ФОРУМ; ИНТРА-М, 2004. С. 6.

[6] См.: *Данилевский Н.Я.* Россия и Европа. М.: Книга, 1991.

[7] См.: *Герасименко В.А.* Защита информации в автоматизированных системах обработки данных: В 2 кн. М.: Энергоатомиздат, 1994.

[8] См.: *Зельнер А.* Байесовские методы в эконометрии. М.: Статистика, 1980.

В.Р. Григорьев, Л.О. Шуркин

СЕТЕЦЕНТРИЧЕСКИЕ ВОЙНЫ С ПОЗИЦИИ СИНЕРГЕТИКИ

На сегодняшний день одной из наиболее актуальных проблем в области обеспечения безопасности государства является исследование вопросов его устойчивости как сложноорганизованной системы к возможным воздействиям со стороны вероятного противника. Предложены новые эффективные методы моделирования информационного противоборства и конфликтных ситуаций различного масштаба и интенсивности с целью определения путей и направлений сдерживания так называемых «сетецентрических войн» (СЦВ), уменьшения риска при принятии решений в кризисных ситуациях.

Ключевые слова: сетевое общество, информационное противоборство, сетецентрические войны, сложная система, устойчивость, синергетика, сложные сети.

Одной из центральных проблем общей теории национальной безопасности является обеспечение адекватной оценки ожидаемых внутренних и внешних угроз и выбора эффективных мер противодействия им в условиях неопределенностей и стохастичности процессов развития сферы информационного противоборства. При этом основное внимание должно быть уделено информационным аспектам проблемы снижения риска неадекватной оценки ситуации и принятия неэффективных и опасных решений, реализация которых может нанести значительный ущерб интересам и безопасности при управлении направленным развитием функциональных систем социально-экономического механизма государства[1]. Эта проблема является фундаментальной как в научном, так и в прикладном планах, и ее решение может оказать существенное влияние на все сферы и объекты безопасности.

Разработки в области информационных технологий резко изменяют взаимодействие как народов, организаций, так и отдельных людей. Быстрое распространение информации ставит под вопрос уместность привычных и обычных организационных и управленческих начал. Военное значение новых организационных наук, которые исследуют сетевые взаимоотношения в противоположность иерархическим моделям управления, также пока еще не полностью понято. Глобализация сетевой связи создает новые уязвимости ключевым национальным информационным инфраструктурам и новые угрозы информационной инфраструктуре исходят именно из достижений в области глобальных телекоммуникаций[2], и как следствие имеют в основе своей сетевую природу.

Общество сейчас переживает распространение сетевой организационной культуры, которая до сих пор не претендовала на роль доминирующей, но тем не менее по эффективности стоит выше нынешней, иерархической. «Кирпичик» иерархической культуры – институт, сетевой – личность. Если институт основан на централизации, вертикальной субординации, штатном расписании и постановке формальных целей, то сетевая организация – на относительной автономии частей, аутсорсинге и распределении рисков. Сетевые сообщества (и террористические, в частности) формируются из личностей, которые несут собственное концептуальное целеполагание, создавая полевые проекты временных виртуальных организаций. Если выдвинутая идея содержит вызов, то заразившиеся ею люди (акторы сетей) собираются в своего рода кластеры (малые миры). Причем один и тот же актор может пребывать одновременно в разных кластерах (сетях). Даже если один кластер сети разрушен, остальные могут продолжать эффективно функционировать. (Применительно к террористическим организациям это означает, что арест даже большой группы террористов может не затронуть работоспособности всей террористической сети.)

Сетевая форма организации позволяет ей быстро адаптироваться к изменяющимся внешним условиям, отмечал один из главных теоретиков сетевого общества Мануэль Кастельс. И, кстати, именно так уже действуют многие крупные транснациональные корпорации, приспосабливаясь к изменчивой геометрии глобальной экономики, легче других выдерживая конкуренцию. Корпорации – образования куда менее инертные, чем бюджетные структуры, к которым относятся и спецслужбы, и системы безопасности в целом. Но теперь, видимо, спецслужбам придется также перенимать сетевые правила игры. Чтобы попытаться победить врага на его же поле, им необходима децентрализация и автономия объединенных общей целью мобильных

адаптивных структур наряду с привычными вертикальными командными линиями. Пока же подверженные директивному планированию спецслужбы просто не успевают отвечать на новые вызовы, и, видимо, в этом кроется одна из причин, почему они не могут быстро перестроиться на борьбу с новыми, нешаблонными видами террористических угроз. Если более продвинутая с точки зрения организационной эволюции сетевая структура ставит задачу атаковать устаревающую иерархическую организацию, то у последней мало шансов выйти победителем. С этой целью, а именно для создания таких эффективных механизмов достижения превосходства над своими противниками в США военно-политическим истеблишментом активно ведутся НИОКР в области борьбы с системами управления, основная цель которых – создание так называемого информационного превосходства непосредственно на поле боя (в том числе виртуального) как «возможности собирать, обрабатывать и передавать непрерывный поток информации при одновременном противодействии противнику».

Террористические акты в США 11 сентября 2001 г. открыли эпоху «мятежевойны» (в ряде трактовок этот термин звучит как «мятежвойна»), наступление которой предсказал еще в начале 60-х годов XX в. русский военный ученый-эмигрант Евгений Месснер. Им были определены принципиальные особенности этого явления: отсутствие линий фронта и четких границ между противниками, превращение общественного сознания в основной объект воздействия, четырехмерное пространство войны (к трем традиционным добавляется информационно-психологическое измерение).

Возникновение сетецентрических (в своей сути сетевых) войн как таковых стало следствием появления асимметричных угроз в современном мире. Согласно определению Института национальных стратегических исследований Национального университета обороны США, под асимметричными угрозами понимаются «использование фактора неожиданности во всех его оперативных и стратегических измерениях, а также использование оружия такими способами, которые не планируются США». Разумеется, данная сугубо американская трактовка применима и в более широком смысле – как использование фактора неожиданности асимметричных действий (операций) против любого государства.

Сегодня очевидно расширение понятия сетевых войн до масштабов глобальной информационной агрессии. Тем более что новый взгляд на угрозы XXI столетия заключается как раз в том, что все чаще основная угроза исходит не от регулярных армий разных стран, а от всевозможных террористических, криминальных и других организаций, участники которых объединены в некие сетевые структуры.

В сущности, эти сетевые организации переводят информационное превосходство в боевую мощь, эффективно связывая интеллектуальные объекты в единое информационное пространство. Происходит трансформация понятия «поле противостояния» в понятие «информационное пространство столкновения интересов». В него включены цели, лежащие в виртуальной сфере: эмоции, восприятие и психика ситуативного противника. Воздействие на новые классы целей достигается за счет тесной интеграции сетевых структур координирующего органа и сетевых структур гражданского общества (например так называемых негосударственных общественных объединений (НГО), вырабатывающих общественное мнение). Поэтому необходимо отметить, что современное противостояние в информационной сфере ведется не только между государствами, но и между негосударственными (неправительственными) организациями и государством.

Это особый, в основном для частных случаев, метод противостояния является необычайно мощным средством как нападения, так и защиты. В нападении сетевые организации, как правило, очень гибки, легкоадаптируемы к различным условиям, универсальны и предоставляют многочисленные возможности взаимодействия. Особенно это характерно для случаев, где субъекты используют тактику *роения*[3].

Как и практически любое новое явление, концепция военного и информационного противоборства и сдерживания пережила определенное смысловое перерождение. О том, что *сетевые войны зачастую становятся не механизмом борьбы с терроризмом и иными асимметричными угрозами* (в качестве которого они изначально рассматривались), *а инструментом самого терроризма и транснациональной преступности*, а также *способом решения определенных политических задач*, говорят многие факты. Межнациональные террористические группы, черный рынок оружия массового поражения, нарко- и иные преступные синдикаты, фундаменталистские и этно-националистические движения, экологические и правозащитные организации, пираты в сфере IT-технологий и иной интеллектуальной собственности, контрабандисты, беженцы и нелегальные мигранты по-прежнему остаются составной частью сетевых войн. Однако к этой же концепции обращаются и революционеры нового поколения, радикалы и другие активисты, начинающие создавать свои идеологии на основе технологических и социокультурных достижений века информации, в которых акценты от отдельного государства смещаются в сторону межнационального уровня глобального гражданского общества. При этом действия всех ука-

занных групп могут носить как национальный, так и транснациональный характер. Естественно, целью некоторых субъектов является уничтожение действующих институтов власти, но целями большинства является подрывная деятельность и дезориентация населения в восприятии истинных и навязанных угроз.

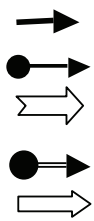
Термин «асимметричная информационная война» (АИВ) занял прочное место и в лексиконе военных специалистов КНР. В настоящее время они разрабатывают концепцию АИВ, которая, как предполагается, будет включать все исторические и национальные представления о том, как воевать на стратегическом, оперативном и тактическом уровне, а также на 36 стратегемах великого полководца и мыслителя Сунь Цзы, который в своих трудах делает акцент на обман, войну знаний и поиск асимметричных преимуществ над противником еще до начала сражения. АИВ определена как «переход от механизированной войны индустриального возраста к войне решений и стиля управления, войне за знания и войне интеллекта»[4].

В рамках проведения информационных операций (ИОп) потенциальный противник посредством использования широкого арсенала информационного оружия может нанести серьезный (а возможно, и невосполнимый) ущерб противнику без использования традиционных контактных силовых методов и не понеся потерь. Воздействуя только на информационно-управляющую систему, он может вывести из строя, обезопасить или практически уничтожить основные элементы гражданской и военной инфраструктуры. Кроме того, атакующий не обнаруживает себя, а атакуемая сторона не всегда в состоянии вовремя и однозначно идентифицировать агрессора. Отсюда следует, что информационная борьба, ведущаяся даже в мирное время, становится реальной угрозой национальной безопасности. Чтобы этого избежать, необходимо обладать сведениями о направлении и средствах реализации возможной угрозы, которые, естественно, охраняются противной стороной со всей тщательностью. Их необходимо добывать и создавать свою систему многорубежной адаптивной защиты в экономической, политической и военной областях. Все это требует новых подходов к разработке, исследованию и развитию средств и методов в области ведения информационного противоборства и крайней его степени – информационной войны.

Глобализирующийся мир требует качественно нового подхода к проблемам принятия решений. Все более настойчиво ставится вопрос о сетевой природе управления: переходе от жестко иерархированной вертикальной системы управления (где четко разделены центральное ядро, в котором и принимаются решения, и периферия,

обязанная эти решения беспрекословно выполнять) к запараллеленной, или распределенной, т. е. сетевой системе, когда полномочия по принятию решений делегируются от центрального ядра к структурным подразделениям. Но в этой ситуации проблема безопасности информации переходит на качественно иной уровень: если в первом случае четкая вертикаль управления позволяет (директивными указаниями) легко обеспечить защиту каналов поступления объективной и достоверной информации, то при сетевой системе опасность утечки информации возрастает на несколько порядков.

Кроме того, при иерархическом и сетевом подходах значительно различаются цели проведения деструктивных управляющих воздействий, направленных на получение контроля над информационными системами, входящих в контур управления ключевыми объектами информационной инфраструктуры страны (рис. 1, 2). На рисунках использованы следующие условные обозначения:



– воздействие на отдельные элементы

– воздействие на связи между элементами

– воздействие на подсистемы

– воздействие на связи между подсистемами

– воздействие на систему

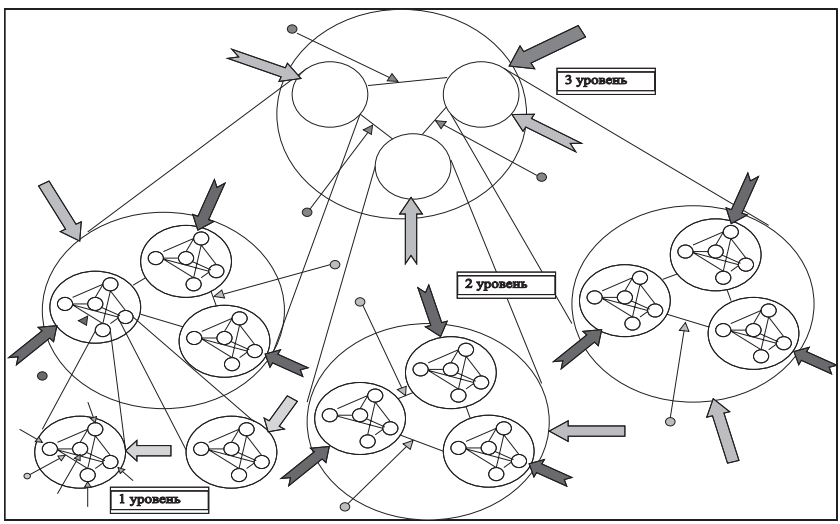


Рис. 1. Иерархическая система управления в условиях ведения ИП

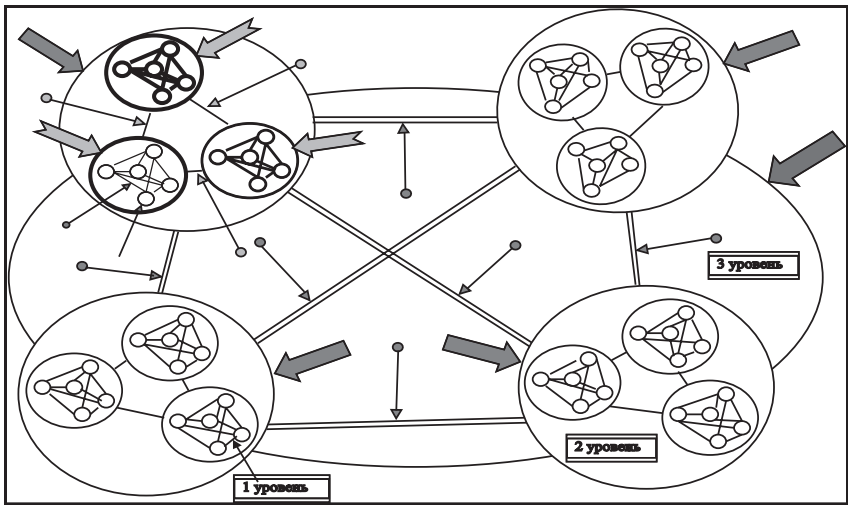


Рис. 2. Сетевая система управления в условиях ведения ИП

В табл. 1 представлены общие задачи информационного противоборства двух динамических систем: нападения и защиты.

Следует отметить, что адекватное понимание проблем информационного противоборства естественным образом вытекает из *общесистемных представлений противоборства* любых двух систем с противоположными интересами и целями, в нашем случае – систем защиты и нападения, каждая из которых представляет собой многофакторную, иерархическую, многоцелевую, сложноорганизованную многоэлементную систему. В этом ракурсе информационное противоборство первоначально может быть рассмотрено исходя из *общей задачи противоборства двух динамических развивающихся сложных систем*.

Динамические системы. Необходимость учета фактора времени при описании сложной системы, а также рассмотрения поведенческих аспектов в движении и развитии сложных систем приводит к необходимости исследования динамической системы.

Под динамической системой S будем понимать сложное математическое понятие:

$S = \langle T, \Phi, \Sigma, \Gamma, X, U, Y, G, R \rangle$,

определяемое следующими аксиомами.

Таблица1

Объекты безопасности / Параметры ИП	Структура	Функции	Информация	Интеллект	Система
Объекты ИП	Структурная схема узлов, элементов и их связей Исполнительные устройства (программно-аппаратные средства)	Сведения о выполняемых функциях Функциональные подсистемы Система управления Сервисы Система защиты	Информация, составляющая гостайну Конфиденциальная информация Информация, составляющая служебную тайну	Экспертные системы, – СУБД и СУБЗ Ситуационные центры Органы управления Лица, принимающие решения	Предназначение системы Архитектура системы Место расположения системы
Предмет ИП	Структурная устойчивость (гомеостазис) Управляемость Качество переходных процессов Надежность Отказоустойчивость Наблюдаемость Идентифицируемость Безопасность	Исполнительные: – обработка; – хранение; – передача управления Вспомогательные: – синхронизация; – маршрутизация и т. д. Защиты (иммунитет)	Целостность Конфиденциальность Доступность Ценность Оперативность Стоимость	Система диагностики Система мониторинга: – обнаружение; – реагирование Система принятия решений Система управления Система обработки знаний: – извлечение знаний; – порождение новых знаний; – синтез знаний	Системы управления Системная устойчивость (системный гомеостазис) Процессы самоорганизации системы: – саморегулирование; – самовосстановление

Продолжение таблицы 1

Объекты безопасности / Параметры ИП	Структура	Функции	Информация	Интеллект	Система
Угрозы	Физическое уничтожение элементов и/или узлов Р/э подавление Техногенные аварии Инициированные сбои и отказы оборудования Нарушение штатных режимов работы Катастрофа (физическое уничтожение подсистемы) Неумышленные: – сбои, отказы оборудования; – ошибки операторов; – природные катаклизмы	Внедрение программно-аппаратных закладок Внедрение вирусов и разрушающих информационных воздействий (РИВ) Внесение недокументированных возможностей Внесение неисправностей в коммутационное оборудование Вывод из строя систем защиты Использование несертифицированных информационных технологий Использование несертифицированных СЗИ	НСД: – перехват; – копирование; – модификация; – подмена Дешифрование Компрометация ключей и СКЗИ Съем по ПЭМИН Отвод «Полицейский режим» Блокирование Хищение Уничтожение	Неадекватность принятия решения в условиях неполноты информации Противоречивости информации Искажения или подмены информации Блокирование Уничтожение	Потеря управления вооруженными силами Потеря управления системами оружия Потеря управления страной Техногенные катастрофы Умышленные: – внесение управляемого хаоса (бифуркация, дезинтеграция); – катастрофа (физическое уничтожение системы) Неумышленные: – сбои, отказы подсистем; – ошибки персонала; – природные катаклизмы

Окончание таблицы 1

Объекты безопасности / Параметры ИП	Структура	Функции	Информация	Интеллект	Система
Способы защиты	Администрирование Контроль за использованием ресурсов Разграничение доступа к ресурсам Организационно-режимные мероприятия	Конфиденциальность доступа Доверенные ОС Биометрическая идентификация пользователей Электронные замки Интеллектуальные карты	Кодирование Пароли Шифрование Имитозащита ЭЦП Защита ключевой информации (квантовая криптография) Стеганография	Аудит Контроль трафика Автоматическая диагностика отказов, сбоев, неисправностей Мониторинг угроз РИВ Мониторинг РИВ Обнаружение и идентификация атакующих информационных воздействий и их источников	Соблюдение политики безопасности: – разграничение доступа к системе; – создание искусственных «иммунных систем»; – поддержание системной устойчивости; – создание «адаптивных распределенных защищенных архитектур»; – разработка минимально необходимой информационной инфраструктуры; – создание систем с «быстрым восстановлением»; – организационно-режимные меры

1. Заданы: множество моментов времени T, макрофункция системы Φ, множество входных воздействий (внешних угроз) Σ, множество возмущений (внутренних угроз) Γ, множество состояний системы X, множество управлений U, множество значений выходных величин Y, структура системы G и отношение эмерджентности R.

2. Множество T есть некоторое упорядоченное подмножество множества вещественных чисел.

3. Макрофункция системы определяется с помощью двух функций:

$S : \Sigma \rightarrow Y$ и $V : \Sigma \times Y \rightarrow C$,

где S – функциональная модель объекта,

V – функция качества, или оценочная функция,

C – множество оценок.

Макрофункция системы определяется парой (S, V).

4. Множество возмущений (внутренних угроз) Γ, или множество неопределенностей, представляет собой множество всевозможных воздействий, которые сказываются на поведении системы. Если такое множество не пусто: $\Gamma \neq 0$, функциональная модель объекта принимает вид $S : \Sigma \times \Gamma \rightarrow Y$, а оценочная функция – $V : \Sigma \times \Gamma \times Y \rightarrow C$.

5. Существует переходная функция состояния системы

$\varphi : T \times T \times X \times \Sigma \rightarrow X$,

значениями которой служат состояния

$x(t) = \varphi(t, \tau, \sigma) \in X$,

в которых оказывается система в момент времени $t \in X$, если в начальный момент $\tau < t$ она находилась в состоянии $x(\tau) \in X$ и в течение отрезка $[\tau, t]$ на нее действовали входные воздействия $\sigma \in \Sigma$.

6. Задано выходное отображение

$\eta : T \times \Sigma \rightarrow Y$,

определяющее выходные величины $y(t) = \eta(t, x(t))$.

Пару (τ, x), где $\tau \in T$, $x \in X$, называют *событием* системы S, а множество $T \times X$ – *пространством состояний* системы.

Конечный набор состояний системы, задаваемый переходной функцией φ и определенный на некотором временном отрезке $[t_1, t_2]$, $t_1, t_2 \in T$, называется *траекторией поведения* системы на интервале $[t_1, t_2]$.

Говоря о движении системы, мы будем иметь в виду траекторию поведения сложноорганизованной системы в фазовом пространстве безопасных состояний.

7. Структура системы G определяется в терминах теории графов: $G = \langle \{S_i\} (S_i, S_j) \rangle$, $i, j = 1, n$; $i \neq j$, где S_i – вершины, (S_i, S_j) – дуги графа.

8. Отношение эмерджентности
$R : \Phi \rightarrow G$.

Данное понятие динамической системы позволяет выработать общую терминологию, уточнить концептуализацию и обеспечить единый подход в рассмотрении приложений, однако является недостаточно конкретным.

В рамках абстрактной теории систем последнее определение дополняется необходимыми доопределениями: конечномерности, линейности, стационарности и др. Задачи, рассматриваемые для динамической системы, традиционны: это вопросы устойчивости, идентификации, инвариантности, наблюдаемости, управляемости и оптимальности, реализуемости и др. Углубленное изучение теории вопроса позволяет грамотно и корректно ставить и решать задачи, связанные с управлением информационной безопасностью сложноорганизованных систем в условиях информационного противоборства.

Принцип холизма. Чтобы эффективно действовать в сложном и нестабильном мире, необходимо принимать во внимание контекст – ближайший и достаточно широкий – изучаемых явлений и событий, т. е. уметь контекстуализировать свои знания. В связи с этим необходимо развивать холистическое видение. «Думай глобально, а действуй локально!» – вот лозунг сегодняшнего дня. Решая задачи информационного противоборства, необходимо понимать способы интеграции и взаимосогласованного, гармоничного развития различных сложноорганизованных социотехнических структур в мире.

Основной принцип холизма, состоящий в утверждении «целое больше суммы частей», может быть прослежен с древних философских учений. Одна из наиболее ранних его формулировок содержится в даосизме, философии Лао-цзы. Однако полный и глубокий смысл этого принципа был выявлен в таких теориях, как гештальтпсихология, теория систем и синергетика.

Принцип рассмотрения от целого к частям, или поведения частей с позиции целого, необычен для классической науки. Последняя движется в ходе анализа, главным образом, от рассмотрения отдельных частей к рассмотрению целого. С синергетической же точки зрения, параметры порядка (характеристики системы как целого) определяют поведение частей (подсистем) сложной системы. Они позволяют существенно редуцировать сложность описания исследуемой системы.

Классический принцип суперпозиции теряет свою силу в сложном и нелинейном мире, в котором мы живем: сумма частных

решений не является здесь решением уравнения. Целое не равно сумме частей. Вообще говоря, оно не больше и не меньше суммы частей. Оно качественно иное по сравнению с частями, которые в него интегрированы. И, кроме того, формирующееся целое видоизменяет части. Коэволюция различных систем означает трансформацию всех подсистем посредством механизмов установления когерентной связи и взаимного согласования параметров их эволюции.

Системный подход предполагает комплексное рассмотрение исследуемого объекта как системы с учетом внутренних и внешних связей и на основе общих принципов сложности и цели. Заметим, что любой объект обладает отличительными особенностями, характеризующими его отдельные стороны (аспекты). Особенности, выделяющие данный объект из совокупности других, являются свойствами этого объекта, которые могут меняться с течением времени, переводя объект из одного состояния в другое.

Фактически выбор того или иного системного представления диктуется удобством решения задач, стоящих перед исследователем или перед научной областью.

Понятие *«система»* обычно связывается с такими понятиями, как *«связь», «структура», «элемент»*. Причем исследователи разных областей вкладывают в эти понятия различный смысл. Однако во всех областях понятие «система» предполагает, с одной стороны, рассмотрение объекта как целого, изучение его внешних параметров и, с другой стороны, некоторую совокупность элементов, связи между которыми образуют структуру[5].

На этапе объединения элементов в систему важнейшую роль следует отвести возникновению коммуникационных связей между элементами, которые через образовавшиеся каналы организуют обмен информацией. Последнее составляет непременное свойство любой сложноорганизованной системы (биологической, социальной, телекоммуникационной).

Применительно к системам исследуемого класса представляет практический интерес интерпретация фазы *бифуркации*, характерной в соответствии с теорией неравновесных процессов по И.Р. Пригожину для любой произвольной системы, имеющей в основе нелинейную динамику образующих ее процессов. Такие системы в своем развитии могут впадать в неравновесные состояния, т. е. состояния потери гомеостазиса. По Пригожину, причинами бифуркаций являются, в первую очередь, флуктуации в системе, в том числе изменения параметров, ведущие к катастрофам[6]. Для кибернетических систем этот перечень можно дополнить эффектом

потери структурной устойчивости не только из-за негрубости системы, но и эффектов нелинейности при входных деструктивных воздействиях. Особо следует выделить влияние информационной недостаточности для обеспечения жизнедеятельности системы: этот фактор практически не изучен.

В большинстве конкретных исследований выбор данного представления объекта как системы в значительной степени определяется выбором исходного расчленения на элементы, так как о связях можно говорить лишь после того, как расчленение произведено и их характер будет определяться типом выделенных элементов.

Возникновение новых задач порождает возникновение новых членений.

Если принять за вероятную основу то, что все искусственные информационно-управляющие системы созданы человеком, с точки зрения выполнения конкретных потребностей, возникающих в процессе решения той или иной целевой задачи, то можно констатировать, что они неразрывно связаны с любой направленной деятельностью, ведущейся в условиях противодействия. И, таким образом, в свою очередь являются предметом борьбы. Наше рассмотрение информационной сферы будет в основном сконцентрировано вокруг различного рода взаимоотношений частей (микроуровня) и целого (макроуровня) – системы.

Мы будем предполагать, что информационная сфера обладает свойствами высокоорганизованной системы – целостностью и единством.

Вместе с тем информационная безопасность может быть рассмотрена как частный случай общего методологического подхода к определению безопасности любой сложной системы, если удается сформулировать это понятие в общесистемном подходе к некоторому достаточно широкому классу систем. В основе представления о безопасности произвольной системы можно рассмотреть следующие основные принципы:

системности – как совокупности взаимосвязанных элементов (субобъектов);

структурности, определяемый только взаимосвязями между элементами системы;

упорядоченности, под которым подразумевается мера соответствия управляемой траектории развития системы поставленным целям; принцип упорядоченности показывает степень приспособленности системы к функционированию во внешней среде, состоящей, как правило, из ряда противоборствующих (конкурирующих) систем (влияний);

надежности, рассматриваемый как мера самосохранения системы; он описывает стремление системы к усовершенствованию своей внутренней структуры, к повышению запаса ее устойчивости;

устойчивости, определяемый как качество, состояние или степень сопротивляемости (адаптивности) системы деструктивным изменениям работоспособности системы, позволяющий восстанавливать в случае негативных последствий последних ранее существовавшее равновесие, т. е. выполнять свои целевые функции в полном объеме;

эффективности, отражающий отношение траекторий, описывающих внешние и внутренние законы поведения систем в условиях противоборства.

Функции *устойчивого,* или *стабильного,* существования произвольной системы определяются следующими существенными факторами:

– возможностью адекватного реагирования на внешние и внутренние угрозы, грозящие ей дезинтеграцией;

– стремлением к укреплению и самоорганизации собственной внутренней интеграции составляющих компонент, т. е. саморазвитием и самосовершенствованием собственной структурной организации;

– стремлением к нахождению в устойчивом фиксированном состоянии стабильности, или покоя, при отсутствии внешних и внутренних возмущений;

– многосвязностью внутренних компонент с возможностью оптимизации необходимого и достаточного для существования системы набора связей, гарантирующего системную устойчивость при любых комбинациях внешних и внутренних деструктивных воздействий;

– возможностью *самовосстановления*, т. е. адаптацией к дезинтегрирующим внешним и внутренним воздействиям и установлением новых состояний устойчивости после выполнения адекватных защитных реакций на деструктивные воздействия (система не должна быть «подвешена» в результате проведенной против нее атаки).

Информационная сфера – это сложная, самоорганизующаяся самореферентная коммуникативная система, обладающая эмерджентными (внезапно появляющимися, неожиданными) свойствами, для описания которой необходимо учитывать теоретические принципы квантовой механики – наблюдаемости и дополнительности.

В сложноорганизованных системах (системах, интуитивно представляемых состоящими из очень большого числа элементов и

их связей, открытых, меняющихся) процессы коммуникации принципиально отличны от процессов в системах с малым количеством элементов, и как следствие их описание требует иного понятийного и методологического аппарата.

Таких понятий как «информация», «обмен информацией», «хранение информации», уже недостаточно для объяснения процессов, происходящих в сложной системе. Центральным понятием для объяснения процессов, происходящих в сети, может быть понятие «*самоорганизация коммуникативного процесса»,* самоорганизация – как «тонкая», сложноорганизованная структура согласованности коммуникаций, когерентное взаимодействие множества информационных объектов, не являющееся следствием какого-то смыслового, целеполагающего управленческого воздействия. По крайней мере, вполне корректным будет предположение о способности или возможности таких систем к сложной самоорганизации.

Синергетическое описание глобальной сети подразумевает наличие как минимум двух уровней рассмотрения – макроуровня, уровня глобальной организации системы, и микроуровня, уровня локальных взаимодействий выделенного элемента (пользователя, сервера). Самым важным качеством синергетических систем является возможность появления новых качеств на макроуровне, которые отсутствуют при рассмотрении деталей на микроуровне.

Современные информационные системы и технологии по своему содержанию и организации все более соответствуют понятию сложной системы[7]. Формирование сложной системы начинается с этапа объединения отдельных элементов и возникновения между ними информационных связей.

Во многом понятие сложной системы воспринимается интуитивно как понятие большой размерности, применяемое к количеству образующих систему элементарных элементов и связей между ними. Если L_i , $i = \overline{1, n}$ – количество связей i-го элемента, то сложность $I = \sum_{i=1}^{n} L_i + 1$. Предельная сложность системы образуется при полносвязной структуре и для n элементов составит $I = n(n-1)+n = n^2$. Связь L_{il} элементов x_i *и* x_j характеризуется своей силой, определяемой как сильная связь для случая $x_i \cap x_j \Rightarrow x_i \cup x_j$ (т. е. для x_i, x_j больше общего, чем различий) и слабая для случая $x_i \cap x_j \Rightarrow \varnothing$ (т. е. x_i и x_j сильно различны). Система становится сложной, если она обладает сильными связями и $I >> I_{kp}$, I_{kp} – критическое значение сложности. В этом случае рождение, жизнь и гибель системы подчиняются законам сложной системы. Одним из них яв-

ляется множественность цели (целевой функции) системы. Можно утверждать, что если отдельный элемент, или сильная связь, есть потенциальная цель, так и любое подмножество элементов и связей системы может быть ее целью. Информационная система как объект защиты (в отличие от какой-либо другой системы) имеет существенную дополнительную сложность, обусловленную тем, что она, во-первых, несамодостаточна, ее цель востребована обслуживаемой ею системой управления, во-вторых, она реализуется в конкретной внешней среде с конечным ресурсом. Информационная система как система сложная имеет некоторую заявленную цель F и множество частных целей f_1, ..., f_m, которые при реализации определенных условий могут противоречить и друг другу, и заявленной цели F. В некоторых случаях это может приводить к самоуничтожению системы. Подобные противоречия целей, как правило, обусловлены ограниченностью ресурсов системы. Таким образом, с точки зрения оказания возможных воздействий на сложную систему необходимо иметь технологии выявления множества частных целей f_1, ..., f_m, используя которые можно управлять развитием всей системы через ее информационное поле.

Ограниченность ресурсов системы также приводит к конфликтам ее элементов за право на ресурс и может порождать противоречия ее целей. Таким образом, внешняя среда информационной системы не только обеспечивает жизнедеятельность системы, но может служить источником ее уничтожения или деградации цели F.

В указанном смысле защитная оболочка системы есть ее способность реализовывать цель F в любых условиях функционирования, в условиях любого противодействия путем образования и востребования противоречивых частных целей. То есть защитная оболочка есть продолжение, усиление цели F системы. Она должна действовать в условиях неопределенности относительно реализующейся цели (F_p) системы, так как определение F_p внутри системы является алгоритмически неразрешимой задачей вследствие несамодостаточности системы.

В свою очередь и злоумышленник вынужден, используя механизмы обучения, преодолевать сложность системы и ее защитной оболочки, реализуя и преследуя интересующую его частную цель. Таким образом, категория «сложность» объединяет в единую метасистему S саму информационную систему, ее внешнюю среду, защитную оболочку и злоумышленника как взаимообуславливающие друг друга элементы с сильными связями. В этой метасистеме S действуют законы сложной системы и перестают действовать привычные процедуры и правила обеспечения безопасности.

Чем обусловлены такого рода кооперативные эффекты в сложных системах? Ведущий процесс в сложной системе – это *самоорганизация*. Нет направляющей руки «оракула», нет программиста. Самоорганизация рождается самой системой в результате потери устойчивости некоего состояния как некоторый, обобщенно понимаемый фазовый переход. Это, пожалуй, самое главное в синергетике. Как известно, сложные системы состоят из очень многих деталей, что порождает возможности очень сложного взаимодействия между этими деталями. Как изучать эти взаимодействия и детали?

Есть два подхода. Во-первых, *редукционизм*, низводящий функционирование системы к микроуровням, деталям. Во-вторых, если так можно сказать, *макрохолизм*, описывающий поведение системы в целом на макроуровне. Не разбирая систему на части и не сводя ее к функционированию на макроуровне, необходимо попытаться понять взаимодействия между микро- и макроуровнем – это первое, и второе (что, пожалуй, самое главное) – отсутствие «направляющей руки» ставит вопрос о сопоставлении между традиционным описанием сложных систем и синергетикой. Единицей описания в традиционном подходе является отдельный элемент рассматриваемой системы, например клетка, нейрон, компьютер в сети. Единица описания в синергетике – это сеть, состоящая из клеток, нейронов, компьютеров. В обычном описании свойства приписываются индивидуальному объекту, в синергетике – ансамблям, множествам объектов. То есть результаты информационного взаимодействия и свойства порождаются не отдельными элементами системы, а их кооперативными отношениями: согласованностью, синхронизацией, когерентностью.

Нет отдельных «специальных» управляющих элементов системы, отвечающих за те или иные ее качества. И если в традиционном подходе описание качеств сложной системы явно или неявно опирается на принцип локализации, то синергетика, как и квантовая механика, существенно нелокальна. Соответственно, в традиционном подходе информация актуально локализована на каких-то носителях, в синергетическом она потенциально распределена. В синергетике нет ничего заранее предопределенного, алгоритмизированного на уровне заранее заданной компьютерной программы, кроме структур и системы, которая при потере устойчивости может рождать какие-то новые вещи.

Описание сложной системы на основе методов самоорганизации дистанцируется от траекторного подхода и соответственно классического детерминизма, создавая новый язык описания со

своими понятиями, новыми ограничениями, налагаемыми на классическую динамику.

Для описания сложности нам надо описать, во-первых, характер начальных условий роста и организации системы, характер связей между компонентами и выяснить, какой параметр является управляющим, ведь параметры могут являться не только функцией времени, но и других факторов (геометрии среды, типов границ, связности компонент и др.), а также управляющих параметров, от которых могут зависеть параметры состояния исследуемой системы.

Встает вопрос о *стратегиях управления сложной системой*. Во-первых, необходимо проанализировать характер неустойчивости системы. Если бы система все время находилась в устойчивом состоянии, то событий в системе не было и параметр времени при описании системы можно было бы не учитывать. Наша задача – описать динамику системы в условиях возможных негативных управляющих воздействий на нее. Для этого мы должны обсудить разные типы ее состояний.

Имеются состояния, которые притягивают к себе (устойчивые «ложбинки», «впадинки», притягивающие множества в пространстве состояний) так называемые *аттракторы,* или паттерны. Память системы – наличие этих аттракторов «ложбинок» в пространстве состояний. Естественно, что система может притягиваться не со всего пространства состояний, а из определенных его областей, так называемых бассейнов притяжения.

Для входа в новое состояние система должна потерять устойчивость. Сначала она была устойчивой (в старом состоянии), потом теряет устойчивость и переходит в новое состояние. За счет чего? За счет случайных колебаний – флуктуаций. Наличие шума – условие перехода из одного устойчивого состояния в другое, но для перехода эти устойчивые состояния должны быть достаточно близки к неустойчивой точке, иначе флуктуации может просто «не хватить», чтобы перекинуть систему из одного состояния в другое.

Таким образом, задача сохранения устойчивости сложной динамической информационной системы в условиях деструктивных информационных воздействий состоит в контроле точек бифуркации, т. е. в областях потери устойчивости, около неустойчивых точек, в окрестностях фазовых переходов.

Это описание указывает на ограничения траекторного подхода (в некоторых случаях траектория становится ненаблюдаемой) со всеми вытекающими отсюда радикальными последствиями в виде пересмотра принципа причинности в сложной системе. Система

становится непредсказуемой не в силу нашего незнания или отсутствия вычислительных мощностей, а в силу таких ее нелокальных качеств, как сложность, нелинейность, открытость, неравновесность, имплицирующих некорректность траекторного описания.

Когда происходит этот переход, то выясняется, что поведение системы описывается совсем не всеми многочисленными компонентами вектора состояния, а гораздо меньшим число параметров, так называемыми параметрами порядка. Если считать систему с бóльшим числом параметров более сложной, а с меньшим – более простой, то можно говорить о том, что в состояниях, близких к фазовому переходу, система упрощается, становится менее сложной, менее хаотической. В этот момент система сама производит сжатие информации – переход от многочисленных параметров состояния к очень немногочисленным параметрам порядка.

Зависимость между параметрами порядка и параметрами состояния не однонаправлена. С одной стороны, компоненты вектора состояния зависят от того, определяется система параметрами порядка или нет. Но есть и обратная зависимость, т. е. векторы состояния влияют на параметры порядка. Такая двухсторонняя зависимость получила у Хакена название круговой причинности[8].

И наконец, важным аспектом самоорганизации является то, что части ведут себя таким образом, что действуют согласованно. Такое поведение можно интерпретировать как консенсус между частями – взаимосогласованность между векторами состояний и параметрами порядка.

Синергетика описывает рождение и формирование сложных систем по сценариям сменяющих друг друга периодов устойчивости и неустойчивости, причем, к примеру, периоды устойчивости могут быть совершенно различными. На начальном этапе (сразу после рождения) система блуждает в пространстве состояний, формируя свой первый аттрактор, первую память, потом в результате деструктивных воздействий или за счет внутренних сбоев и поломок перескакивает за счет потери устойчивости и флуктуаций в другую область и формирует аттрактор там, потом может пойти «искать третий аттрактор или перескочить в первый, и так далее. За счет этого формируется рельеф состояний сложной системы: области устойчивости, особые точки, туннели по переходу из одной области в другую. Образуется понятие *цели самоорганизации сложной системы*.

Очевидно, что существует необходимость в рассмотрении вопросов *структурно-методологической формализации сферы ИП* на основе общесистемного подхода к анализу отношений базисных

качеств систем защиты и нападения в рамках их прогрессирующего информационного противоборства.

Далее с изложенных выше позиций предложен теоретико-множественный подход к описанию проблемы безопасности информационных систем в условиях деструктивных воздействий на них из внешней среды и инициализированных внутренних возмущений.

Теоретико-множественный подход к описанию проблемы безопасности информационных систем в условиях деструктивных воздействий из внешней среды и инициализированных внутренних возмущений

Теоретико-множественное определение системы: система есть собственное подмножество $X_S \subset X$, где X – прямое (декартово) произведение множеств X_i, $i = \overline{1, n}$:

$X = X_1 \times X_2 \times X_3 \times ... \times X_n$.

Декартовым произведением множеств называется множество конечных наборов элементов $(x_1, x_1, ..., x_n)$, таких, что

$x_1 \in X_1, x_2 \in X_2, ..., x_n \in X_n$.

Каждый элемент $x_i \in X_i$ в свою очередь может быть множеством, что позволяет описывать иерархию достаточно сложных систем, в том числе систем обеспечения информационной безопасности на государственном уровне.

Принято различать два аспекта безопасности[9]. С одной стороны, безопасным называют явление и / или состояние какого-либо носителя опасности, которое не содержит угрозы и / или возможного вреда для его окружения. С другой стороны, свойство безопасности приписывают объекту, надежно защищенному от опасных для него воздействий. Таким образом, понятие безопасности имеет две функции: *внутреннюю,* характеризующую свойства сопротивляемости объекта по отношению к действиям среды, и *внешнюю,* определяющую воздействие объекта на среду. Определим эти понятия[10].

Внутренняя безопасность *есть критерий целостности системы или показатель ее гомеостаза. Иначе, безопасность характеризует способность системы поддерживать свое нормальное функционирование в условиях воздействия среды и внутренних возмущений.*

Под внешней безопасностью будем понимать *способность системы взаимодействовать со средой без нарушения гомеостаза последней. Иначе, воздействие системы на среду не приводит к необратимым изменениям или нарушениям важнейших параметров, характеризующих состояние среды, принятое за допустимое.*

Соответственно будем различать и постановки задач по исследованию внешней и внутренней функций безопасности: в 1-м случае основное внимание уделяется динамике среды в условиях воздействий со стороны обследуемой системы, а во 2-м наибольший интерес представляет поведение системы в активной среде.

Строго говоря, в указанной постановке речь идет о *взаимодействии двух систем – среды и собственно системы*, образующих некую метасистему противоборства. Только при одновременном изучении их в масштабе метасистемы противоборства можно получить полное представление о взаимном влиянии обеих систем друг на друга и оценить это влияние с позиции обеспечения безопасности. Однако ставить такую задачу практически невозможно ввиду ее громоздкости, о чем упоминалось выше. Следовательно, для практики исследования свойств безопасности важно уметь методически корректно разделять систему и среду.

Структура и характер связей между элементарными системами зависят от свойств среды окружающего пространства и механизма обмена информацией. Поэтому важным объектом при описании множества взаимосвязанных систем (коллектива) является структура всех допустимых каналов обмена между системами. Эта структура по существу является объемлющим пространством, внутри которого реализуются различные структуры связи.

В соответствии с работой[11] дадим теоретико-множественное описание проблемы обеспечения безопасности, рассматривая ее как конкретизацию общей задачи взаимодействия (противоборства) двух динамических систем.

Итак, имеем две системы – собственно систему $Sys^{(1)}$, динамика которой характеризуется соотношением (1), и среду – $Sys^{(2)}$ (рис. 3).

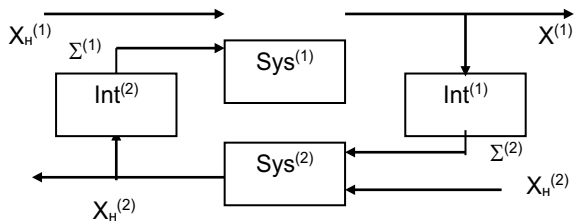


Рис. 3. Схема взаимодействий двух динамических систем

Оператор каждой из взаимодействующих противоборствующих систем (объект + система управления), находящейся в некото-

ром начальном состоянии $X_н$ и испытывающей действие внутренних возмущений (внутренних угроз) Γ (в том числе – структурных), внешних воздействий (внешних угроз) S, в состав которых можно включить и управление безопасностью, можно записать в виде

$$Sys^{(1)}: T \times X_н \times \Gamma \times \Sigma \Rightarrow X, i = 1, 2 \quad (1)$$

Взаимодействие (противоборство) систем проявляется через воздействие 1-й на 2-ю, и наоборот, а эти связи охарактеризуем соответствующими операторами

$$Int^{(1)}: T \times X^{(1)} \Rightarrow \Sigma^{(2)}; \; Int^{(2)}: T \times X^{(2)} \Rightarrow \Sigma^{(1)} \quad (2)$$

где (*Int* = Interaction) описывают зависимость внешнего для другой системы воздействия от состояния первой. Структурную схему взаимодействий (противоборства) представим графически (рис. 4).

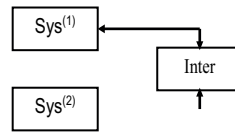


Рис. 4. Структура и механизм взаимодействия систем

Как известно, в кибернетике широко применяется ряд показателей, описывающих динамические свойства системы. Кроме того, используется другая группа характеристик, которая отражает качество технической реализации системы: надежность, массу, габариты, энергопотребление и др. Однако безопасность пока не принято рассматривать как специфическое свойство динамических систем, что, на наш взгляд, существенно обедняет их исследование.

Безопасность – есть комплексный критерий оценки качества любой системы, характеризующий как динамику системы, так и техническое воплощение системы.

Особо оговорим, что *безопасность есть показатель качества системы как целостности;* его нельзя отнести к какому-либо элементу системы, для характеристики последнего можно использовать лишь частные критерии, например, надежность. Если же говорят о безопасности фрагмента какой-либо системы, то, строго говоря, молчаливо исходят из трактовки его как системы.

ГОСТом, а также в практике разработки и эксплуатации технических систем выделяют различные характеристики потери устойчивости работы системы: катастрофы, аварии, неисправности. Мы же будем рассматривать опасность разрушения социотехнической системы, потерю ею своего функционального предназначения без

учета вредных последствий для персонала. Именно такое понимание безопасности позволяет применить его не только к анализу искусственных систем, но и для анализа естественных систем, организационных и др.

Объективным показателем интегральной безопасности системы является ее безопасное состояние в процессе функционирования. Поэтому о безопасности системы можно судить по ее динамике, которая характеризуется известными показателями: устойчивостью, качеством переходных процессов, управляемостью, наблюдаемостью, идентифицируемостью. Нарушение этих характеристик приводит или к разрушению системы, или к невыполнению ею своей задачи, или как минимум к ухудшению качества управления. Следует отметить, что эти показатели для систем с развитой динамикой редко выражаются аналитически, а обычно имеют вид условий, способствующих выполнению соответствующего требования, или в общем случае оценки указанных характеристик системы базируются на результатах математического моделирования.

Более того, фундаментальные показатели качества системы, строго говоря, являются пороговыми характеристиками – система может быть или устойчивой (или управляемой, наблюдаемой, идентифицируемой), или нет. Подобный подход к безопасности является малоинформативным: желательно количественно оценить степень опасности (безопасности) системы.

В связи с этим необходимо коснуться одного методологически важного вопроса. Иногда высказывается утверждение, что безопасность системы определяется через надежность ее работы. Однако это не так. Надежность системы есть показатель ее способности сохранять свои наиболее существенные свойства (безотказность, ремонтопригодность и др.) на заданном уровне в течение фиксированного промежутка времени при определенных условиях эксплуатации. Надежность определяется вероятностными показателями, характеризующими реакцию системы на отказ – событие, заключающееся в нарушении работоспособности системы из-за изменений ее параметров, внезапных или постепенных. В качестве показателя надежности обычно используют вероятность безотказной работы или наработку на отказ (среднее время безотказной работы).

Механизм применения теории надежности в управляемых системах состоит в следующем. Известны статистические характеристики выхода из строя элементов и определен показатель надежности системы, представляющий собой функцию, которая описывает работоспособность системы при отказах. Проблема состоит в установлении связи между характеристиками элементов и функцией –

показателем надежности. Эта зависимость позволяет пересчитать исходные данные в результирующий статистический критерий.

Итак, методы теории надежности обычно применяются для анализа режима эксплуатации конструктивно оформленной системы, состоящей из ненадежных элементов. В основе теории надежности лежит событие как некоторый одноразовый акт, позволяющий в случае многократных повторений (!) определить вероятность его последствий. Теория безопасности не может исходить только из многократности явлений, имеющих опасные последствия: для гибели системы достаточно создания одной катастрофической ситуации. Кроме того, в рамках теории надежности трудно оценить многоальтернативность поведения системы при отказе. Важно еще и то, что в основе безопасности систем лежит необходимость наблюдать за динамическими процессами, а не контролировать отдельные события в системе – ведь за каждым событием (например отказом) стоит процесс, ведущий, возможно, к опасному результату. Наконец, надежность слабо коррелирует с целостностью системы. Разумеется, это не исключает использования различных статистических оценок при анализе безопасности.

Отсюда следует, что теория безопасности методологически шире теории надежности, последняя может использоваться для исследования отдельных сторон безопасности систем, особенно эффективно-искусственных.

Итак, имеем динамическую систему, находящуюся под действием внешних *управлений* (*внешних угроз*) и внутренних *возмущений (внутренних угроз*). Весь этот спектр внешних и внутренних воздействий может при определенных условиях привести к разрушению системы.

Общая задача состоит в построении оценки, позволяющей в процессе работы системы численно определить угрозу распада системы, чтобы своевременно принять меры по его недопущению.

Очевидно, что такая оценка, вообще говоря, должна быть построена на траекторных движениях системы, т.е. представлять функционал

$J_б = J_б (t, x, u, \sigma, \gamma), x \in X, u \in U, \sigma \in \Sigma, \gamma \in \Gamma,$

где X – множество возможных состояний системы, U – множество допустимых управлений, Σ – множество внешних воздействий (внешних угроз), Γ – множество внутренних возмущений (внутренних угроз).

Тогда все пространство состояний системы можно разделить на две области: одна будет составлять множество опасных для существования системы состояний X_0, а другой будут принадлежать

все безопасные состояния $X_б$. В сумме эти множества опишут все возможные состояния системы $X = X_0 \cup X_б$. Тогда наша первоначальная задача заключается в построении множества безопасных состояний $X_б$. Но выделить множество – значит найти его границу $Г_б$, которая и будет нести информацию о безопасности. Тогда *показатель безопасности* $J_б$ *есть мера удаления текущего состояния системы* **x** *от границы* $Г_б$, что иллюстрирует рис. 5.

Полезно выделить две противоречивые тенденции, с которыми приходится сталкиваться при построении $X_б$. С одной стороны, чтобы гарантировать работоспособность системы, из этого множества требуется исключить все режимы, которые бы приводили к ее деструкции. Значит, множество допустимых с точки зрения безопасности состояний системы следует по возможности сужать. Но ограничение множества допустимых состояний стесняет возможности функционирования системы, а следовательно, уменьшает область достижимости и целевое множество. Преодоление противоречия осуществляется поиском компромисса.

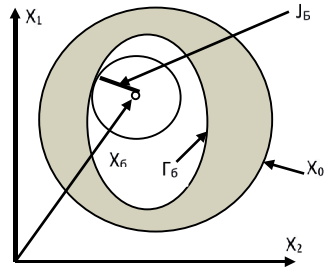


Рис. 5. Показатель безопасности в пространстве состояний системы

Строго говоря, граница области безопасных состояний может быть получена путем наблюдения за системой или на основе полномасштабного математического моделирования ее работы. В последнем случае производится перебор всех подозрительных воздействий выделенного класса на систему и проверяется реакция на них системы. Это позволяет выделить катастрофические состояния, а значит найти $Г_б$. Однако такой подход не может быть положен в основу создания системы борьбы с угрозами, так как он обладает принципиальной задержкой в принятии решения: факт аварийности режима устанавливается лишь после того как он произошел, что исключает прогноз и проведение защитных мероприятий.

Выход из тупика следует искать в удалении от границы безопасности, точнее, уменьшении области безопасности, ее модификации ($X_{бм}$ и соответственно $\Gamma_{бм}$). Действительно, наличие некоторого «запаса» безопасности предоставит (системе, ЛПР) время на парирование угроз и повысит уровень защищенности системы. Дело лишь в том, какие соображения положить в основу выбора такого запаса, как его обоснованно назначить. Точно выбрать величину запаса, как уже указывалось, можно по результатам моделирования и выработке научно обоснованных норм, предотвращающих разрушительные процессы при всех условиях работы системы. Однако сделать это далеко не всегда удается. Действительно, многие системы не поддаются математическому моделированию в силу их сложности, отсутствия пригодного математического аппарата или невозможности предугадать характер воздействий. Тогда на помощь приходит опыт общения с подобными системами, здравый смысл и интуиция.

Однако, даже если построена модифицированная граница области безопасности $\Gamma_{бм}$, то находить в пространстве S кратчайшее расстояние от текущего состояния системы, задаваемого вектором x, до границы $\Gamma_{бм}$ – дело многотрудное. Итак,

во-первых, наличие модифицированной области безопасности $X_{бм}$ в пространстве состояний наиболее объективно свидетельствует об удаленности текущего режима работы системы от состояния, угрожающего ее целостности. Однако для повышения временного ресурса, для устранения неполадок в системе, для увеличения оперативности и качества управления было бы *желательно располагать информацией о причинах,* обусловливающих приближение состояния системы к опасной границе. Для этого необходимо рассмотреть факторы, определяющие появление опасных для системы режимов, т. е. *требуется проанализировать угрозы,* проникающие через единственный канал – через воздействия на систему;

во-вторых, желательно найти $J_б$, имеющие бóльшую физическую наглядность и мéньшую сложность вычисления, нежели определение расстояний в X.

Идею решения 1-й задачи подсказывает рис. 5: если раньше $\Gamma_{бм}$ строилась на основе информации о состоянии (выход модели), то теперь мы хотим привлечь для этого сведения о входных воздействиях (σ, γ, u). Основную проблему при построении оценки безопасности доставляют трудноизмеряемые параметрические возмущения $p \in \Gamma$ (кроме параметрических мы различаем еще структурные возмущения) и воздействия внешней среды γ Учет остальных факторов или чрезвычайно затруднен (структурные

возмущения в естественных системах), или прост до тривиальности (эффект управления искусственными системами).

Будем исходить из предположения, что область безопасности $X_{бм}$ удалось построить. Тогда *принципиально задача состоит в пересчете этого подпространства пространства состояний в пространства входных воздействий – параметрических* $X^{п}_{бм}$ *и внешних* $X^{в}_{бм}$ *возмущений.*

Формально на основе (1) можно записать

$$Sys^{-1}: T \times X_{бм} \Rightarrow \{\Sigma, \Gamma\}. \qquad (3)$$

Однако такое решение затруднительно, так как из реакций системы трудно однозначно осуществить идентификацию воздействий, выделить причинную обусловленность динамики, т. е. установить вклад каждого возмущения в результат – состояние. Поэтому приходится обходиться без процедуры общего пересчета (3) и по отдельности строить области безопасности для каждого входного воздействия:

$$C_{p}^{-1} : T \times X^{п}_{бм} \Rightarrow \Gamma; \qquad (4)$$

$$C_{\Sigma}^{-1} : T \times X^{в}_{бм} \Rightarrow \Sigma. \qquad (5)$$

Методически смысл построения заключается в нахождении соответствия границы $\Gamma_{бм}$ множества $X_{бм}$ границам в пространствах параметров и воздействий среды – соответственно $\Gamma^{П}_{бм}$ и $\Gamma^{в}_{бм}$. Техника процедуры состоит в определении методами моделирования соответствующих предельных значений указанных величин, превышение которых грозит разрушением системы. А именно, перебирается весь спектр воздействий и находится реакция системы на каждый входной сигнал. Те сигналы, которые приводят к распаду системы, и признаются опасными. Разумеется, метод перебора (например, метод Монте-Карло) применяется только тогда, когда входные и выходные воздействия не удается связать аналитически.

Сложность технологии усугубляется еще одним обстоятельством: в общем случае для динамических нелинейных систем существует зависимость области нормального функционирования системы от параметрических и внешних возмущений. Грубо говоря, для каждого уровня внешних воздействий имеется свое множество допустимых значений параметров системы. Это утверждение иллюстрируется рис. 6.

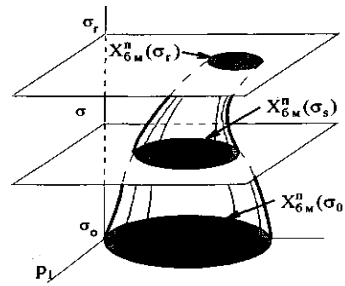


Рис. 6. Деформация параметрической области безопасности при различных возмущениях

Горизонтальная плоскость рисунка есть множество параметров $P = \{p_i\} \in \Gamma$, $i = 1, 2$, где выделена область безопасности $S^{П}_{бм}$. По ординате отложена величина уровня внешних возмущений σ с тремя конкретными значениями – σ_0, σ_s, σ_r. Для разных уровней возмущений область $S^{П}_{бм}$ меняется, т. е. становится их функцией (можно предположить, что по мере роста воздействий на систему параметрическая область безопасности сужается).

Таким образом, в результате построений мы располагаем двумя наборами *взаимосвязанных* множеств: областями безопасности $S^{в}_{бм}$ и $S^{П}_{бм}$, построенными в пространствах входных воздействий и флуктуирующих параметров соответственно. Тем самым *при оценке безопасности можно перейти от изучения состояний системы к наблюдению за входными сигналами, а значит заменить анализ следствия анализом причин.*

В принципе, как пояснялось выше, предложенную технологию, базирующуюся на обратных преобразованиях (4), (5), следует оценить позитивно, так как процедура позволяет обратиться к истокам процесса нарушения безопасности, а значит, дает возможность более обоснованно подойти к построению системы и разрешает увеличить время, необходимое для принятия решения по предотвращению критической ситуации. Но существуют, конечно, и негативные стороны:

1. Обращение непосредственно к угрозам, исходящим от среды и нарушений в системе, привело *к размножению* областей безопасности. Вместо итоговой области в пространстве состояний мы вынуждены иметь дело с несколькими областями, по числу каналов проникновения угроз в систему, да к тому же объединенными функциональными связями. Конечно, это сделает алгоритмы обеспечения безопасности более громоздкими.

2. Известная трудная доступность измерениям воздействий среды явно усложнит построение системы предотвращения угроз. Гораздо проще иметь дело с результирующей информацией о состоянии системы.

Наконец, следует отметить, что в приведенную схему трудно вписывается процедура идентификации, столь успешно применяемая в задачах типа «черного ящика». Действительно, для восстановления входной информации по наблюдениям за выходными процессами необходимо определенное время анализировать реализацию сигнала, что иногда исключается при решении задач безопасности, требующих оперативного вмешательства в работу системы; запаздывание в этом может грозить уничтожением системы.

Известно, чтобы построить работоспособную систему, надо удовлетворить, кроме требования устойчивости, условиям управляемости, наблюдаемости и идентифицируемости. Как правило, эти требования очевидны, и их выполнение не вызывает принципиальных затруднений. В правильно спроектированной системе можно избежать их нарушений, а значит, угроз существованию системы. Ресурсные требования обычно удовлетворяются на стадии проектирования или подготовки системы к выполнению конкретной задачи. Превышение их порогового значения может привести к недостижению функциональной цели, к ухудшению экономических показателей, к уменьшению эффективности работы системы. Однако таким ущербом часто пренебрегают, так как он очень мал по сравнению с угрозой разрушения системы.

Таким образом, *в качестве критериев безопасности можно рекомендовать показатели устойчивости системы и количество ресурсов, необходимое для выполнения задачи. Именно эти оценки целесообразно положить в основу практического построения областей безопасности*. Кроме того, важным показателем системы безопасности является ее стоимость, функционально связанная с системой.

Итак, для оценки безопасности предлагается перейти от функционала, вычисляемого на движениях (траекториях) системы, к анализу устойчивости системы при изменении входных воздействий и параметрических возмущениях. Роль дисциплинирующих показателей играют количество ресурса, необходимого для функционирования системы, и стоимость мер поддержания безопасности на требуемом уровне.

Ранее показывалось, что управление безопасностью основывается на широком математическом моделировании поведения си-

стемы в ожидаемых условиях ее функционирования. Для реальных систем нельзя предположить, что требуемый уровень безопасности будет достигнут на основе только аналитического исследования.

В первую очередь моделирование должно предоставить данные о модифицированных областях безопасности в пространстве внешних воздействий $X^{в}_{бм}$ и параметров $X^{П}_{бм}$, точнее, об их границах – $S^{в}_{бм}$ и $S^{П}_{бм}$ – предлагаемая технология изложена выше. Затем в пространстве возмущений F, $f = \{\sigma, \rho\} \in F$ критерий можно выразить через метрику этого пространства

$$J_{16} = \min \rho \, \| f - \Gamma_{бМ}(f) \|. \qquad (6)$$

Представляется возможным записать оценку безопасности как величину отклонения текущего возмущения от некоторого, признанного за оптимальное значение:

$$J_{26} = \rho \, \| f - f_{орt} \|. \qquad (7)$$

Наконец, можно дискретизировать задачу оценки, когда за опасное состояние принимается случай превышения показателя некоторого допустимого значения

$$J_{36} = \rho \geq \{\rho_{доп}\}. \qquad (8)$$

Тем самым вокруг состояния f формируется некоторая область размером $P - P_{доп}$, выход за которую должен заставить систему реагировать. Аналогично можно за аварийное состояние признать факт его принадлежности границе:

$$J_{46} = \{f \in \Gamma_{бм}\}. \qquad (9)$$

Приведенные оценки без каких-либо изменений можно использовать в статистических постановках. Следует лишь отметить, что частота измерений должна удовлетворять требованию теоремы В.А. Котельникова, что позволит избежать запаздывания в идентификации критического состояния.

Указанные критерии должны играть роль входной информации для системы управления безопасностью.

Упомянутые выше понятия «траектория», «равновесные точки», их классификации и т. п. являются понятиями из теории динамических систем, общими для динамических систем любой природы.

Вклад указанных составляющих модели в устойчивость конечной структуры можно исследовать в рамках следующих взаимодополнительных задач.

1. Для заданного набора элементарных динамических систем определить существование и особенности структуры связи, повышающей стабильность всей системы.

2. Для заданной структуры связи определить существование и особенности множества элементарных систем, для которых указанная структура связи оказывает стабилизирующее действие на всю систему.

Таким образом, первая задача в основном направлена на описание свойств элементарных систем по отношению к возможности их стабилизации структурами обмена, а вторая задача подчеркивает свойства некоторой сети способствовать стабилизации набора элементарных систем достаточно широкой природы.

В частности, первая задача позволяет поставить вопрос о существовании для данного набора элементарных систем хотя бы одной сети (в заданной топологии), повышающей стабильность данного набора. Возникает соблазнительная задача классификации динамических систем на «хорошие», которые стабилизируются хотя бы одной связывающей сетью, и «плохие», не стабилизирующиеся ни одной сетью из допустимого набора. Содержательные оценки типа «хорошие – плохие» могут быть заменены на противоположные, но качественные отличия таких систем очевидны и интересны.

Перечисленные особенности определяют специфику проблемы исследования, ее принципиальную новизну и необходимость поиска новых подходов к решению этой проблемы и связанных с ней задач.

Выводы

1. Анализ метасистемы S как сложной системы показывает, что для рассматриваемых систем существуют только две возможности реализации защиты от деструктивных управляющих воздействий:

– существенное увеличение их естественной сложности до уровня, исключающего возможность их изучения злоумышленником за приемлемое время;

– реализация защитной оболочки в виде механизмов обучения (в отдельных случаях самообучения) с последующим накоплением и обобщением базы знаний о штатно работающей системе в условиях специально организованного ее корректного функционирования.

На практике одновременно реализуются оба подхода со стратегией максимального потребления полезной информации на собственное обучение с одновременным обеспечением ее минимальной доступности потенциальному злоумышленнику.

2. На структурном уровне глобальная конфронтация в настоящее время происходит между иерархическими (государственные игроки) и сетевыми структурами (негосударственные игроки). Будучи гораздо более избыточной и гибкой, последняя структура обладает большей устойчивостью, что затрудняет ее окончательное разрушение.

3. С синергетической точки зрения, одним из основных подходов к решению глобальных проблем является смена императива: *не политика силового давления и выкручивания рук, а поиск способов коэволюции сложных социальных и геополитических систем*. Осуществление политики силовыми методами слишком опасно в современном сложном, нелинейно развивающемся мире, где даже случайные сбои в разветвленных информационных, компьютерных сетях могут привести к мировой катастрофе. Чем сложнее организована и более многофункциональна система, тем она более неустойчива. Поэтому понимание форм совместной жизни разнородных, находящихся на разном уровне развития социальных и геополитических структур, путей их устойчивого коэволюционного развития становится конструктивной альтернативой сегодняшнего дня.

4. Предложено использовать в качестве критериев безопасности показатели устойчивости системы и количество ресурсов, необходимое для выполнения задачи. Именно эти оценки целесообразно положить в основу практического построения областей безопасности. Кроме того, важным показателем системы безопасности является ее стоимость, функционально связанная с системой.

Примечания

[1] См.: *Черешкин Д.С., Кононов А.А.* Комплексная оценка безопасности сложных информационных систем // Мир компьютеров и мир людей – взаимодействие и конфликты: Сб. тр. междунар. симп. Кишинев, 1999. С. 24–25; См.: *Черешкин Д.С., Кононов А.А., Будин О.А.* Экспертная система оценки риска нарушения информационной безопасности для систем управления информационной безопасностью // Информатизация правоохранительных систем: Сб. тр. IX Международной конф., 7–8 июня. 2000 г. М., 2000. 545 с.

[2] См.: *Аузан В., Африн Д.* Технологии против сетей // Эксперт. 2001. № 38 (298).

[3] *Роевой интеллект* (англ. Swarm intelligence) описывает коллективное поведение децентрализованной самоорганизующейся системы. Рассматривается в теории искусственного интеллекта как метод оптимизации. Системы роевого интеллекта, как правило, состоят из множества агентов, локально взаимодействующих между собой и с окружающей средой. Сами агенты обычно довольно просты, но все вместе, локально взаимодействуя, создают так называемый роевой интеллект.

[4] См.: *Гриняев С.* Концепция ведения информационной войны в некоторых странах мира // Зарубежное военное обозрение. 2002. № 2.

[5] См.: *Фельдбаум А.А.* Основы теории оптимальных автоматических систем. М.: Физматгиз, 1963.

[6] См.: *Пригожин И.* От существующего к возникающему. М., 1985. 327 с.

[7] *Капица С.П., Курдюмов С.П., Малинецкий Г.Г.* Синергетика и прогнозы будущего. М.: Наука, 1997. С. 87.

[8] См.: *Хакен Г.* Синергетика. Иерархии неустойчивостей в самоорганизующихся системах и устройствах. М.: Мир, 1985.

[9] См.: *Могилевский В.Д.* Формализация динамических систем. М.: Вузовская книга. 1999.

[10] См.: Там же.

[11] См.: Там же.

С.Т. Петров, А.А. Тарасов

ЦИФРОВОЕ НАСЛЕДИЕ КУЛЬТУРЫ: ПРОБЛЕМЫ ФОРМИРОВАНИЯ, РАЗВИТИЯ И БЕЗОПАСНОСТИ

Рассматриваются гуманитарные и технологические проблемы формирования, сохранения, обеспечения доступа и другие аспекты информационной безопасности цифрового наследия культуры. Предложен процессный подход к описанию объекта цифрового наследия как модели объекта культуры и к цифровому наследию как сложной социотехнической системы. Описаны и проанализированы информационные системы по видам культурных ценностей и типам культурной деятельности, а также содержание угроз информационной безопасности в сфере культуры.

Ключевые слова: цифровое наследие, культурное наследие, социотехническая система, информационная система, информационная безопасность.

В современном мире проблемы развития культуры неразрывно связаны со становлением информационного общества[1], а вопросы сохранения и доступа к культурному наследию – с вопросами обеспечения национальной безопасности в информационной сфере[2]. Одним из основных показателей уровня развития в России информационного общества, ее успеха на международной арене становится достоверность, целостность, сохранность и доступность культурно-исторического наследия в цифровой форме.

Стратегия и Программа развития информационного общества в Российской Федерации на 2011–2020 гг. определяют чрезвычайно высокие показатели в области цифрового контента. По сути, поставлена задача формирования в кратчайшие сроки беспрецедентных по объему и составу цифровых информационных активов[3].

Самая значимая часть таких активов, состоящая из «уникальных ресурсов человеческих знаний и форм выражения»[4], представляет собой цифровое наследие культуры (ЦНК).

Феномен ЦНК является чрезвычайно сложным для описания и понимания явлением, находящимся на стыке информационного, гуманитарного и естественно-научного знания. Между тем междисциплинарный характер ЦНК и его значимость не является очевидным, а само ЦНК часто понимается лишь как совокупность цифровых копий настоящих культурных ценностей, причем обычно считается, что сами эти цифровые копии ничем особым не выделяются среди других информационных активов. Именно поэтому в настоящее время цифровое наследие играет вспомогательную роль в культуре и жизни общества. Доля квалифицированно оцифрованных, организованных и должным образом хранящихся объектов культуры в цифровой форме весьма мала как относительно самого культурного наследия, так и относительно информационных активов в целом. Также недооценивается пока и потенциально широкая востребованность ЦНК. Между тем потенциал использования ЦНК в самой культуре, образовании, науке, досуге чрезвычайно велик. Вероятно, ЦНК будет играть основополагающую роль в сохранении аппарата традиции человечества – коллективной памяти, а также являться важным механизмом сохранения индивидуальной, семейной и родовой памяти.

В рамках развития информационного общества можно сказать, что переход к цифровому миру и цифровому наследию является закономерным, направленным процессом эволюции коллективной памяти – ее социальных институтов и технических средств. Данный процесс уже привел нас к цифровой ноосфере. При этом субъектами жизнедеятельности в этой новой среде обитания являются не только люди, но и все более активные и автономные компьютерные системы.

Обеспечение информационной безопасности ЦНК и создание системы управления его безопасностью – новая комплексная проблема, которую необходимо решать, опираясь как на традиционные практики обеспечения безопасности культурного наследия, так и на методы и средства обеспечения безопасности информационной сферы.

Цель настоящей статьи – рассмотрение задач описания, формирования, хранения и использования ЦНК с учетом различных аспектов его информационной безопасности.

***

Описание и ценностная оценка цифрового наследия – информационно-культурного фундамента нового общества имеет не только философские, аксиологические аспекты, но и представляет сугубо практический интерес, связанный с оценкой значимости цифрового наследия как информационного актива, например, в задачах ИБ.

В последние годы в методологии науки, в различных областях знаний, проектировании информационных систем начинает применяться аппарат веерных матриц[5]. Появление данного средства исследования связано прежде всего с широким применением методов информационных наук в различных предметных областях. Веерные матрицы используются для описания и анализа предметных областей философии, биологии, управления и др., а также для построения поисковых алгоритмов.

ЦНК, являясь объектом междисциплинарных исследований, нуждается в описании, которое может служить основой для фундаментальных и прикладных исследований самого цифрового наследия, управления им и его безопасностью, построения связанных с ним информационных систем.

В этой связи применение веерных матриц позволяет, в частности:

– представить онтологии цифрового наследия как области научного исследования;

– служить удобным средством описания иерархий объектов и критериев, используемых, например, в методе анализа иерархий (МАИ);

– ранжировать угрозы информационной безопасности ЦНК;

– строить и оптимизировать поисковые алгоритмы, используемые при решении задач в области цифрового наследия.

В качестве достаточно условного примера рассмотрим веерную матрицу (табл. 1) отношений взглядов специалистов в различных областях на такие понятия, как «культура памяти», «институты памяти» и «информационно-коммуникационные системы». Эти отношения (понятия) являются элементами матрицы.

Таблица 1

Веерная матрица отношений между взглядами различных специалистов на организацию коллективной памяти

Уровни Организации	Специалисты		
	Культурологи памяти 1	Сотрудники учреждений культуры 2	Специалисты по ИКТ 3
Культура памяти 1	**Культура коллективной памяти**	Культурное наследие	Цифровое наследие
Институты памяти 2	Механизмы коллективной памяти	**Учреждения культуры**	Информационная инфраструктура сферы культуры
Информационно-коммуникационные системы 3	Культурология медиа	Информационные ресурсы сферы культуры	**ИКТ сферы культуры**

Веерные матрицы позволяют наглядно представлять как предметную область в целом, так и отдельные аспекты деятельности в предметной области. Деятельность по сохранению культуры и культурных ценностей, осуществляемая различными специалистами в более узких областях, чем приведенные в табл. 1, представлены веерной матрицей (табл. 2).

Таблица 2

Веерная матрица деятельности по обеспечению сохранности культурного наследия

Уровни иерархии	Специалисты		
	Культурологи таксонов 1	Хранители фондов 2	Специалисты по ИБ 3
Уровень объектов 1	**Сохранение признаков и смыслов объектов**	Сохранение объектов	Целостность файлов
Уровень сегментов 2	Сохранение культур	**Сохранение фондов**	Целостность баз данных
Уровень наследия 3	Сохранение культурного кода	Сохранение культурного наследия	**Целостность цифрового наследия**

В дальнейшем необходимо формализовать действия над веерными матрицами, включая их рекурсивное развертывание (автомодельность), и описать с помощью таких матриц и действий над ними иерархию сущностей и явлений, связанных с ЦНК.

***

Каждый объект культуры является уникальной моделью мира[6], а цифровой образ объекта культуры является уникальной моделью такой модели, полученной в результате деятельности трофических цепей, в которых осуществляются аналогово-аналоговые, аналогово-цифровые, цифро-аналоговые и цифро-цифровые преобразования. Объект ЦНК – результат множества сложных, взаимодействующих процессов, которые могут описываться функциями, переводящими модели из одного состояния в другое. Некоторые такие преобразования и состояния могут считаться недопустимыми, например не обеспечивающими адекватность цифрового образа исходному объекту или ведущими к нарушению целостности ЦНК.

Конечно, модели моделей объектов культуры появились задолго до цифровых копий. Это происходило как в рамках простого копирования текстов в целях распространения, так и в более сложных вариантах, таких как инсценировка пьес. Производные модели сыграли свою роль и в области проблем безопасности культурных ценностей. Была признана необходимость использования достижений науки и техники для сохранения культурного наследия, использования замещающих средств (муляжи, фотографии) в случае утраты самих объектов. Были сформулированы и принципы, определяющие обеспечение безопасности культурных ценностей как в военное, так и в мирное время. ЦНК является особой формой накопления информации и важнейшим информационным активом общества.

Процессы создания объектов культуры как моделей эволюционировали и в рамках отдельных типов культурной деятельности (например, совершенствование техники живописи), и в плане использования технических новинок (фотография), становясь все динамичнее (кино). При этом непрерывно возрастало техническое оснащение культуры – от сложнейших технологий производства аудиоаппаратуры до превращения крупных музеев в высокооснащенные и защищенные объекты, сравнимые с предприятиями оборонного комплекса.

При увеличении возможностей информационно-коммуникационных технологий (ИКТ) цифровые модели начинают приобретать все большую сложность и собственные новые черты и возможности, не сводимые к исходной модели. Само ЦНК является сложной социотехнической системой, включающей взаимодействующие цифровые объекты (комплексы объектов ЦНК), подсистемы их создания, хранения и использования, а также группы людей, обслуживающих или использующих ЦНК.

Жизненный цикл объекта ЦНК включает фазы подготовки и создания объекта; его эксплуатацию (хранение и использование, включая копирование); вывод из оборота и консервацию; остаточное использование (копий или частей объекта); утилизацию. Жизненный цикл ЦНК как сложной системы включает жизненные циклы объектов ЦНК, технологий и инфраструктуры, организаций, а также нормативно-правовой базы в области ЦНК.

Объект ЦНК может устаревать. Данный процесс может быть связан с физической деградацией носителя, утратой информационных свойств, например целостности, а также потерей соответствия исходному объекту. Последнее может быть связано с изменением исходного объекта, а также появлением новых средств оцифровки, позволяющих отображать исходный объект более адекватно. Проблема повторных оцифровок весьма неоднозначна, в частности в силу того, что процесс оцифровки может оказывать негативное воздействие на оригинал.

В основе ЦНК лежат мастер-копии (эталоны, цифровые оригиналы). Пока общепринятого определения объекта ЦНК не существует. Как правило, под мастер-копией понимают самый ранний по времени создания цифровой объект в конкретном технологическом цикле, помещенный на хранение, имеющий таковой статус и существующий в каждый момент времени в единственном экземпляре. Общий вид технологического цикла любого объекта ЦНК имеет вид «аналог-цифра-аналог».

С 1960-х годов начались компьютеризация некоторых учреждений культуры и попытки использования методов точных наук для анализа культурных процессов, произведений искусства и автоматизации творческой деятельности[7]. Первые цифровые копии произведений искусства были примитивны (кто-то, может быть, помнит портрет Джоконды, состоящий из черточек и крестиков), но иногда эти копии обладали важными полезными свойствами (скажем, распечатать роман М. Булгакова «Мастер и Маргарита» можно было без опечаток, купюр и потенциально неограниченное число раз).

Сама сущность цифровых методов и прогресс компьютерной техники привели к тому, что:

– цифровые технологии начинают превосходить по качеству и стоимости все остальные способы создания образов объектов культуры;

– цифровые образы могут безошибочно передаваться и копироваться;

– цифровые образы и/или их элементы могут быть связаны между собой различными семантическими связями;

– совокупность цифровых образов может рассматриваться как единое цифровое культурно-историческое пространство;

– цифровые образы могут сохраняться потенциально неограниченное время.

При этом существует значительное число технических, организационных и психологических проблем, пока не позволяющих цифровым технологиям и образам вытеснить традиционные методы и сами оригиналы из активного оборота.

Новые технологии дают возможность придавать цифровым образам новые свойства и качества (например, реконструкции более раннего или позднего состояния, изображенного на картине), комбинировать реальные и виртуальные объекты (дополненная реальность) и др.

Исследования в области искусственного интеллекта и возрастание мощности вычислительной техники вплотную приблизили создание динамических моделей личности авторов и персонажей, способных к активному общению с читателями и зрителями, а также между собой.

Такие тенденции в области ЦНК соответствуют глобальным тенденциям развития «нового цифрового мира»[8], иногда даже определяя и опережая их.

Не заглядывая в будущее, приведем краткий обзор основных технологий и систем, применяемых в сфере культуры. В документе «Основы законодательства Российской Федерации о культуре» от 9 октября 1992 г. № 3612-I в редакции, действующей с 1 января 2014 г., введены:

– основные понятия сферы культуры, прежде всего дефиниция «культурные ценности», что определяет особенности создаваемых информационных ресурсов и систем в сфере культуры;

– права и обязанности в сфере культуры, что определяет цель и предназначение информационных систем;

– типология культурной деятельности и виды учреждений культуры, что определяет функциональное назначение информационных систем;

– организация управления культурой, что определяет облик автоматизированной системы управления отрасли.

Место ЦНК как информационной системы в ряду информационных систем в сфере культуры представлено в табл. 3.

Таблица 3

Назначение информационных систем в сфере культуры

Основные системы
Обеспечение творческого процесса
Формирование фондов
Учет фондов
Сохранность фондов
Экспонирование фондов
Обслуживание посетителей
Билеты, читательские билеты
Экскурсионное обслуживание
Лекционное обслуживание
Обеспечивающие системы
Охранно-пожарные системы
Инженерно-климатические системы
Связь
Управляющие системы
Принятие решений
Мониторинг сферы культуры
Статистическая отчетность
Документооборот
Финансово-хозяйственная деятельность
Кадры
Информационная безопасность
Дополнительная деятельность
Наука
Образование
Издательская деятельность
Коммерческая деятельность

Окончание таблицы 3

Распространение произведений
Трансляции
Тиражирование
Демонстрации
Цифровое наследие, порталы и сайты
Подготовка цифровых материалов
Хранение
Доступ
Управление контентом
Сервисы
Финансово-учетная система
Внешние системы
Государственные системы
Системы общего пользования

Информационные ресурсы и сервисы, входящие в ЦНК, должны:

- в полноте и целостности отражать культурное наследие (реестры, каталоги);
- быть аутентичными и достоверными;
- репрезентативно представлять отечественную и мировую культуру;
- учитывать возможности и ограничения различных групп пользователей;
- быть взаимосвязанными с государственными и негосударственными информационными ресурсами.

Многообразию культурных ценностей, зафиксированных в Основах законодательства Российской Федерации о культуре, соответствует многообразие способов их фиксации, сохранения, восприятия, трансляции. Основные онтологии (описания), ресурсы и системы по видам культурных ценностей приведены в табл. 4.

Таблица 4

Информационные описания, ресурсы и системы по видам культурных ценностей

Культурные ценности	**Виды ресурсов и функции систем**
Нравственные идеалы	Этические и информационные онтологии
Эстетические идеалы	Критерии отбора; цифровые образы
Нормы и образцы поведения	Фиксация типовых и маргинальных образцов поведения документальными и художественными средствами
Языки, диалекты и говоры	Электронные словари, корпуса текстов, библиотеки, поисковые системы; аудиоархивы; диалектные геоинформационные системы (ГИС); дистанционное обучение
Национальные традиции и обычаи	Этнографические базы данных, аудио- и видеофиксация, трансляции, дистанционное обучение
Исторические топонимы	Исторические и современные базы топонимов; ГИС
Фольклор	Фиксация текстов, аудио и видео; системы классификации
Художественные промыслы и ремесла	Описание и видеофиксация; системы классификации
Произведения культуры и искусства	Цифровые образы, описания и метаданные; системы классификации и поиска
Здания и сооружения	Цифровые чертежи; фотографии; видеоматериалы; 3D-модели и реконструкции; средства мониторинга
Предметы	Цифровые образы, описания и метаданные; системы классификации и поиска
Технологии	Описание и видеофиксация технологических процессов, результатов деятельности, использования
Территории	ГИС; фотографии; видеоматериалы; 3D-модели и реконструкции; мониторинг
Объекты	Планы, фотографии; видеоматериалы; 3D-модели и реконструкции; мониторинг

При таком широком спектре и разнице в масштабах типов культурных ценностей ясно, что ЦНК не сводится к совокупности образов отдельных предметов культуры. При этом остается открытым вопрос, что такое, например, целостное цифровое представление такого явления, как язык, или такого объекта, как музей-заповедник.

Культурная деятельность характеризуется множеством типов деятельности, часть из которых отражена в действующем законодательстве. Функции программно-технических средств и систем по обеспечению данных типов деятельности для создания соответствующих объектов и сервисов ЦНК представлены в табл. 5.

Таблица 5

Функции информационных средств и систем по типам культурной деятельности

	Тип деятельности	**Системы и средства**
1	Профессиональное и любительское творчество	Системы подготовки, обработки, редактирования материалов; средства фиксации и представления результатов творчества; средства коллективной работы; социальные сети
2	Создание и распространение аудиовизуальной продукции	Системы записи, монтажа, специальных эффектов; средства тиражирования; средства трансляции, воспроизведения, визуализации; сети распространения
3	Издательская деятельность	Системы верстки; доставка обязательного экземпляра; реестры СМИ; книжные летописи; мониторинговые и аналитические системы
4	Культурно-досуговая деятельность	Средства представления культурных ценностей и подготовки контента; базы данных в области досуга и развлечений; компьютерные игры
5	Эстетическое воспитание граждан	Системы подготовки материалов, визуализации; дистанционное обучение
6	Образование в сфере культуры и искусства	Системы дистанционного обучения
7	Научно-исследовательская деятельность в сфере культуры и искусства	Базы данных, электронные архивы и библиотеки; поисковые системы; информационно-аналитические системы; средства коммуникации
8	Сохранение, использование, популяризация и государственная охрана культурного наследия	Системы хранения и доступа к цифровому наследию, порталы и сайты в сфере культуры; автоматизированные системы охраны и обеспечения безопасности культурных ценностей
9	Музейная деятельность	Системы автоматизации; средства оцифровки; электронные каталоги и реестры; порталы
10	Библиотечная деятельность	Системы автоматизации; средства оцифровки и распознавания текстов; электронные каталоги и библиотеки; порталы
11	Архивное дело	Системы автоматизации; средства оцифровки; электронные описи, путеводители, архивы; порталы; СЭД; СХД
12	Деятельность по изготовлению и реставрации художественных и декоративных изделий	Средства моделирования; средства мониторинга; 3D-печать

Окончание таблицы 5

	Тип деятельности	**Системы и средства**
13	Деятельность в сфере декоративно-прикладного искусства, дизайна, архитектуры	Системы автоматизации проектирования
14	Культурный туризм	Средства организации экскурсий и туристического обслуживания; краеведческие базы данных
15	Информационная деятельность, направленная на популяризацию сферы культуры	Средства мониторинга сферы культуры; средства доставки информации; сайты и порталы; социальные сети
16	Иные виды культурной деятельности: зоопарки	Базы данных: биоразнообразие, Красная книга; мониторинг животных

Проблемы больших данных, оптимизации хранения и повышения гарантоспособности систем хранения и доступа, новые формы представления и анализа контента – вот далеко не полный список задач, которые стоят перед сферой культуры в области ИКТ. При этом стратегической задачей является создание доверенной среды, включающей меры по повышению доверия к субъектам культурной деятельности, информационным ресурсам, технологиям, инфраструктуре.

Новые свойства активных диалоговых моделей авторов и персонажей произведений, виртуальные экскурсоводы, дополненная реальность – поле научных проблем и инновационных разработок в области ЦНК.

***

Сфера культуры – культурные ценности, учреждения культуры, система управления, информационные активы подвергаются многочисленным угрозам и рискам внутреннего и внешнего характера. Содержание угроз безопасности сообщений, относящихся к культурным ценностям, заключается в возможности дезорганизации и разрушения системы накопления и сохранения документов, составляющих культурные ценности. По мере роста масштаба и значимости цифрового наследия резко возрастают угрозы нарушения его информационной безопасности: целостности, доступности, а также правомерного использования. Для демонстрации способности государства и учреждений культуры защищать свои инфор-

мационные интересы и цифровое наследие настоятельно требуется создание полномасштабной системы информационной безопасности.

Проблемная ситуация в области информационной безопасности сферы культуры[9] характеризуется, в частности, следующим:

– отсутствует концепция и стратегия информационной безопасности сферы культуры;

– отсутствует нормативно-правовая база информационной безопасности;

– не определены модели нарушителя информационной безопасности;

– не проранжированы угрозы информационной безопасности;

– не проанализированы и не оценены риски информационной безопасности;

– не определена политика информационной безопасности на ведомственном и организационном уровне.

В сфере культуры можно выделить три области обеспечения безопасности информационных активов:

– обеспечение информационной безопасности активов, влияющих на безопасность культурного наследия;

– обеспечение безопасности цифрового наследия как части культурного наследия;

– обеспечение безопасности информационных активов сферы культуры как отрасли народного хозяйства.

Значимость объектов ЦНК связана с теми объектами культуры, образами которых они являются[10]. Однако эта зависимость не является однозначной. В сфере культуры существуют многочисленные способы и методики ранжирования, определения значимости объектов и их совокупностей. В последнее время прежде всего в интересах информационной безопасности начинают формироваться методики оценки информационных активов как произвольной природы, так и в различных областях жизнедеятельности, например, в банковской системе Российской Федерации.

На оценку значимости объекта ЦНК влияют качество оцифровки, а также процедура проведения (кто проводит, контроль качества и пр.). В свою очередь выбор параметров оцифровки зависит от оценки значимости объекта оцифровки. Определение методов, параметров и сроков оцифровки зависит не только от значимости объекта оцифровки, но и от текущего состояния объекта, перспектив сохранности и др.

Оценка значимости информационных систем и ресурсов влияет в том числе:

- на политику сохранения информационных ресурсов;
- на организацию доступа;
- инвестиционную и страховую политику.

Оценка качества информационных активов определяется, в том числе:

- составом;
- достоверностью и аутентичностью;
- качеством оцифровки.

Оценка объемов информационных активов определяется:

- правовым статусом активов;
- качеством активов[11];
- количеством единиц хранения;
- объемами информации в компьютерных единицах измерения.

На оценку значимости информационных ресурсов влияет и оценка носителей, на которых они находятся, а также информационных систем, в которых эти активы хранятся и обрабатываются.

Инвентаризация и оценка значимости информационных активов – первый шаг к обеспечению информационной безопасности ЦНК.

Идентификацию, классификацию и ранжирование угроз удобно проводить в форме веерных матриц. В силу недостатка места приведем одну из таблиц (табл. 6), описывающих содержание угроз ИБ.

Таблица 6

Содержание угроз информационной безопасности в сфере культуры

Объекты угроз	**Содержание угроз**
Культура	Утрата идентичности и целостности
Культурное разнообразие	Уменьшение разнообразия состава документов
Культурное сообщество	Ослабление связей, утрата культурных коммуникаций
Культурная деятельность	Нарушение прав, дезорганизация бизнес-процессов
Культурные ценности	Дезорганизация и разрушение системы накопления и сохранения документов, составляющих культурные ценности, изменение системы оценок; нарушение доступности
Культурное наследие	Нарушение принципов и системы оценки значимости
Нематериальное культурное наследие	Искажение и утрата контекста, средств выражения и трансляции ценностей; нарушение культурных коммуникаций

Окончание таблицы 6

Объекты угроз	**Содержание угроз**
Объект культурного наследия	Нарушение целостности, изменение, утрата документов
Национальное культурное достояние	Нарушение системы оценок, системы управления организациями, индивидуальных и коллективных творческих процессов, приоритетов оцифровки
Культурные блага	Нарушение доступа к услугам и культурным ценностям
Культурное пространство	Изменение границ пространства, разрыв связей
Государственная культурная политика	Отсутствие, несанкционированное изменение документов, определяющих политику, средств проведения политики
Творческая деятельность	Нарушение прав и условий по созданию культурных ценностей и их интерпретации; нарушение конфиденциальности
Организация культуры	Дезорганизация деятельности организаций
Права собственности	Изменение и фальсификация документов; неправомерное использование документов
Полномочия ФОИВ	Дезорганизация системы управления, информационных потоков
Целевые программы	Ложные цели, нарушение координации, дезорганизация деятельности исполнителей
Система учета	Дезорганизация учета, фальсификация учетных данных, неправомерное использование результатов учета
Система мониторинга	Дезорганизация системы мониторинга, фальсификация данных мониторинга
Виды деятельности	Дезорганизация бизнес-процессов по видам деятельности
Культурные фонды	Нарушение целостности, изменение прав собственности
Сохранность культурного наследия	Нарушение мер охраны и технологий обеспечения сохранности

***

Сфера культуры представляет сложную систему, сочетающую черты централизованной и сетевой организации. Культура оказывает значительное влияние на большинство сторон поведения личности, функционирование общества, деятельность органов власти, международные отношения. В условиях становления информационного общества сфера культуры становится областью особых и значительных угроз и рисков национальной самобытности, традиционным укладам, суверенитету.

Цифровое наследие культуры – один из значимых информационных активов нации, общества, государства. Недооценка потен-

циала данного актива может привести к негативным последствиям для самой культуры, образования, науки. ЦНК формируется в результате эволюции институтов коллективной памяти и информационно-коммуникационных технологий. Информационная основа ЦНК – взаимосвязанные модели (образы) объектов культуры. Данные модели получаются в результате комплекса процессов, многократно преобразующих информацию из аналоговой формы в цифровую и обратно. Многообразие видов культурных ценностей и типов деятельности в сфере культуры, учреждений культуры информационно-коммуникационных технологий и систем, различных групп авторов, специалистов и пользователей ведет к появлению чрезвычайно сложной социотехнической системы – цифрового наследия культуры.

Формирование, гарантированное сохранение и эффективное использование ЦНК возможно только на основе системного анализа социокультурных и технологических процессов, создания системы информационной безопасности ЦНК.

Примечания

[1] См.: *Белл Д.* Грядущее постиндустриальное общество / Пер. с англ. Изд. 2-е. М.: Academia, 2004. 482 с.

[2] См.: *Стрельцов А.А.* Правовое обеспечение информационной безопасности России: теоретические и методические основы. Минск: Беллітфонд, 2005. 304 с.

[3] Так к 2015 г. планировалось достичь следующих показателей: доля архивных фондов, включая фонды аудио- и видеоархивов, переведенных в электронную форму, – не менее 20 %; доля библиотечных фондов, переведенных в электронную форму, в общем объеме фондов общедоступных библиотек – не менее 50 %, в том числе библиотечных каталогов – 100 %; доля электронных каталогов в общем объеме каталогов Музейного фонда Российской Федерации – 100 %.

[4] Согласование национальных подходов к сохранению цифрового наследия: Пер. с англ. И.Н. Андреевой, Н.Б. Богдановой, Е.А. Губиной, Д.Е. Осадчук М.: МЦБС, 2013. 360 с.

[5] См.: *Кордонский С.Г.* Классификация и ранжирование угроз // Отечественные записки. 2013. № 2 (53). С. 52–73.

[6] См.: *Прангишвили И.В.* Системный подход и общесистемные закономерности. М.: Синтег, 2000. 528 с.

[7] См.: *Моль А.* Социодинамика культуры: Пер. с фр. 2-е изд. М.: URSS, 2005. 416 с.

[8] См.: *Шмидт Э., Коэн Дж.* Новый цифровой мир. Как технологии меняют жизнь людей, модели бизнеса и понятие государств: Пер. с англ. Сергея Филина. М.: Манн, Иванов и Фарбер, 2013. 368 с.

[9] См.: *Кондратьев Д.В., Ненашев А.Н., Петров С.Т., Тарасов А.А.* Проблемы сохранения цифрового культурного наследия в контексте информационной безопасности // Вестник РГГУ. Серия «Информатика. Защита информации. Математика». 2013. № 14 (115). С. 36–52.

[10] См.: *Петров С.Т., Тарасов А.А.* Обеспечение безопасности информационных активов в сфере культуры // Современные проблемы и задачи обеспечения информационной безопасности: Труды Всероссийской научно-практической конференции «СИБ-2014». М.: МФЮА, 2014. С. 57–64.

[11] См.: *Юмашева Ю.Ю.* Нормативно-методическое регулирование процессов оцифровки – обязательная составляющая цифровизации культурного наследия // Справочник руководителя учреждений культуры. 2013. № 7. С. 4–14.

А.Е. Сатунина, Л.А. Сысоева

МЕТОДОЛОГИЧЕСКИЕ АСПЕКТЫ ФОРМИРОВАНИЯ СИСТЕМЫ МЕТРИК ПРИ РЕАЛИЗАЦИИ ПРОЕКТА ИНФОРМАЦИОННОЙ СИСТЕМЫ

В статье рассматриваются методологии, используемые в процессе реализации проекта по созданию сервис-ориентированной информационной системы и влияющие на формирование ее метрик. Реализуемые в настоящее время проекты ИС характеризуются тем, что еще на начальных этапах проекта требуется предусматривать способы и методы измерения необходимых показателей и метрик, которые будут востребованы при интеграции реализуемой системы с BI-системой. В результате проведенного анализа была сформирована многоуровневая структура метрик информационной системы сервис-ориентированной архитектуры, разработанная с учетом методологий RUP, ITIL, ITSM, BPM, проектного менеджмента.

Ключевые слова: процессный подход; управление проектом ИС; сервис-ориентированная архитектура, управление сервис-ориентированной информационной системой; метрики информационной системы.

В современных условиях одним из факторов, оказывающим влияние на эффективность функционирования организаций, является уровень применения систем аналитической обработки информации для поддержки принятия решений на основе массивов первичных данных, накапливаемых в корпоративных приложениях. Практически все компании-вендоры программного обеспечения в сфере корпоративных ИС и хранилищ данных (IBM, Oracle, Microsoft, SAP и т. д.) включают в свой портфель программные средства BI-системы. Востребованность и развитие

технологии интеллектуального анализа бизнес-информации привели к появлению нового направления BI 2.0, которое связано с проактивным анализом имеющейся в организации информации в текущий момент времени.

Другой аспект, влияющий на активизацию использования систем аналитической обработки информации, связан с широким внедрением процессного управления (BPM)[1], где большую роль играет аналитическая информация по показателям результативности бизнес-процессов.

Востребованность аналитических систем в информационной архитектуре предприятия/организации вызывает необходимость предусматривания методов и моделей интеграции существующих и вновь разрабатываемых программных приложений еще на этапе реализации проекта ИС.

Поэтому цель данной статьи – попытаться определить методологии и подходы к формированию системы метрик, которые должны учитываться в ходе осуществления проекта по созданию информационной системы при дальнейшей интеграции ее с системами аналитической обработки информации.

Процессный подход к управлению проектом ИС

В основе управления проектом ИС лежит процессный подход, являющийся одним из восьми принципов менеджмента качества, который декларирует, что «желаемый результат достигается эффективнее, когда деятельностью и соответствующими ресурсами управляют как процессом»[2]. В стандартах по менеджменту качества «любая деятельность, в которой используются ресурсы для преобразования входов в выходы, может рассматриваться как процесс»[3].

При организации управления проектом ИС под процессом будем понимать структурированную совокупность взаимосвязанных и взаимодействующих видов работ, выполняемых в определенный интервал времени, преобразующих входы в выходы и направленных на достижение конкретного результата. Будем учитывать, что: 1) процесс имеет один или несколько входов и преобразует их в определенные, запланированные выходы; 2) процесс включает средства контроля и управления, необходимые для надежного предоставления выхода, удовлетворяющего заданным требованиям.

Принятое определение согласуется с понятием процессного подхода. «Систематическое определение и менеджмент процессов,

применяемых организацией, и особенно взаимодействие этих процессов могут рассматриваться как процессный подход»[4].

Поскольку процесс всегда является целенаправленным, он должен включать анализ требований и формирование целей, мониторинг, контроль и оценку текущего состояния процесса, выработку корректирующих действий в случае отклонения показателей от допустимых значений. Поэтому любой процесс включает две составляющих: непосредственно создание результата (производство выхода) и управление созданием результата (производством выхода)[5]. Согласно ITIL управление процессом – это деятельность по планированию и упорядочиванию процесса с целью его эффективного, результативного и согласованного выполнения[6].

Важность применения процессного подхода в управлении заключается в структуризации различных видов деятельности. Структурированные процессы позволяют управлять качеством в соответствии с циклом PDCA (рис. 1), что обеспечивает достижение стабильных результатов, соответствующих установленным нормам и требованиям, при рациональном расходовании ресурсов.

В соответствии с методологией PDCA на этапе контроля осуществляется сбор информации о ходе реализации процесса, определяются показатели процесса, в том числе KPI, выявляются отклонения показателей от пороговых значений и проводится их анализ с выявлением причин отклонений.

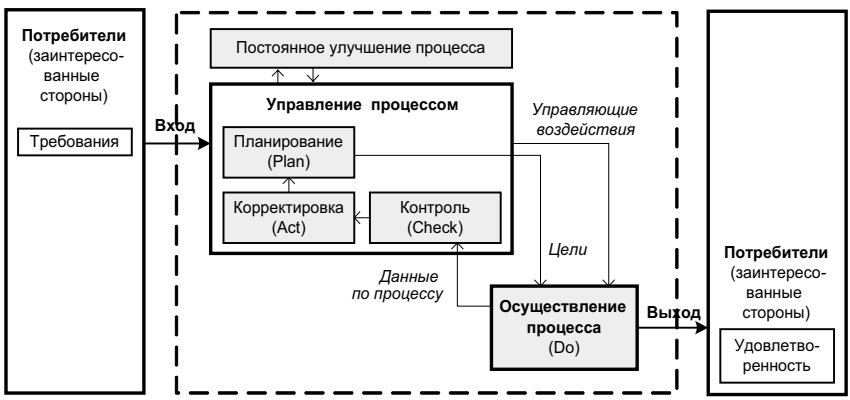


Рис. 1. Модель организации процесса, согласуемая с циклом PDCA

Методология процессного подхода применяется и в проектной деятельности. При выполнении работ по проекту ИС выделяют процессы: создание проектного продукта (в данном случае ИС) и управление процессом его создания, т. е. управление проектом (рис. 2).

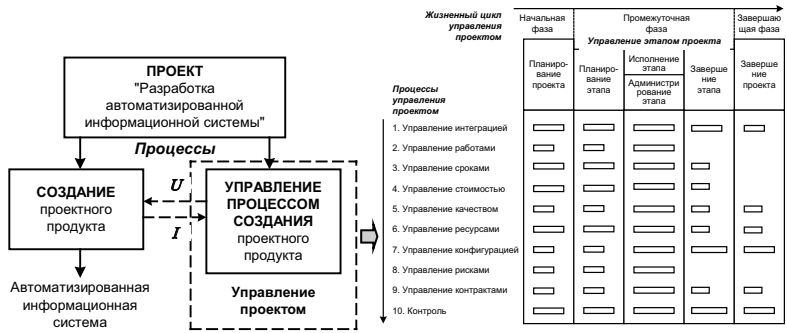


Рис. 2. Структура процессов при реализации проекта ИС

Взаимосвязи между процессами управления и фазами ЖЦ проекта представлены на рисунке 2. Связи между процессами предприятия и процессами проекта ИС регламентируются стандартом ГОСТ Р ИСО/МЭК 15288-2005[7].

Методология создания проектного продукта – информационной системы

Наиболее распространенной методологией, в которой инструментально поддерживаются все этапы жизненного цикла создания ИС, является методология Rational Unified Process (RUP), в основе которого лежит итерационный, пошаговый подход.

RUP – это унифицированный, четко определенный процесс, описывающий структуру жизненного цикла проекта, роль и ответственность отдельных исполнителей, выполняемые ими задачи и используемые в процессе разработки модели, отчеты, документацию, средства контроля за ходом и качеством разработки и механизмы тестирования систем[8].

В методологии RUP выделяются основные процессы (моделирование бизнес-процессов; управление требованиями; анализ и проектирование; реализация; тестирование; развертывание) и вспомогательные процессы (управление изменениями; управление проектом; управление средой). Структура процессов проекта ИС, осуществляемого на основе методологии RUP, представлена на рисунке 3.

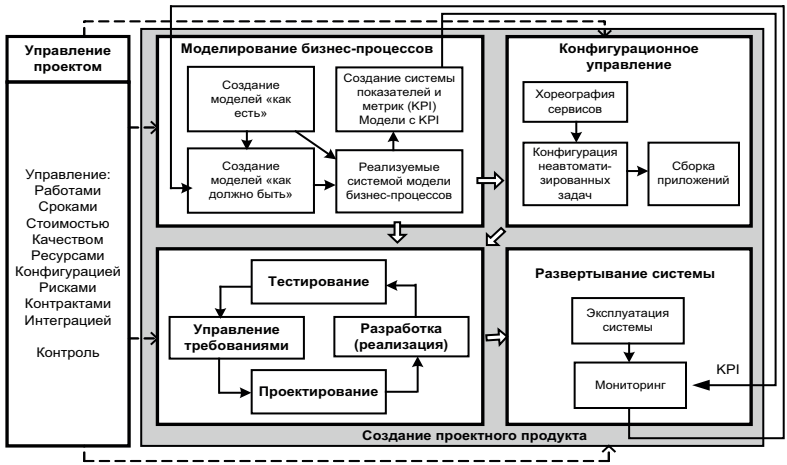


Рис. 3. Создание ИС с использованием методологии RUP

Как видно из схемы, одной из составляющих процесса моделирования является создание системы метрик и показателей бизнес-процессов. Кроме того, модели бизнес-процессов разрабатываются с определением KPI, которые учитываются при мониторинге показателей ИС. Результаты мониторинга и анализа уровня отклонения показателей от пороговых значений передаются в модуль моделирования для совершенствования или оптимизации моделей бизнес-процессов.

Сервис-ориентированный подход к архитектуре ИС

При проектировании архитектуры ИС применяются различные подходы, одним из которых является сервисный подход. Архи-

тектура сервис-ориентированной ИС рассматривается как многоуровневая структура (рис. 4):

1) инфраструктура (программно-аппаратная платформа как совокупность ИТ-сервисов);
2) компоненты (функциональные компоненты приложений);
3) сервисы (атомарные прикладные функции, реализующие логику бизнес-процесса);
4) бизнес-процессы (модели бизнес-процессов и сборка сервисов в соответствии с логикой бизнес-процесса);
5) бизнес-сервисы (сервисы, предоставляемые потребителям);
6) интеграция (обеспечение межуровневого взаимодействия).

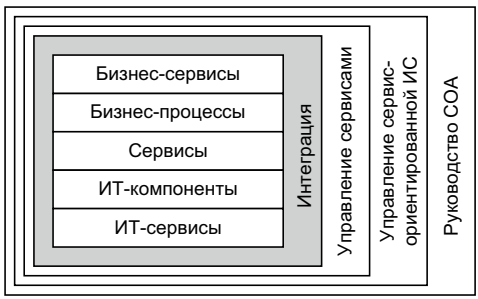


Рис. 4. СОА как многоуровневая структура

На каждом уровне СОА проводятся измерения с применением соответствующих наборов метрик, в состав которых входят метрики бизнес-сервисов, бизнес-процессов, сервисов, компонент приложений, ИТ-сервисов.

В качестве ИТ-сервисов понимаются «услуги ИТ: совокупность функциональных возможностей информационных и, возможно, неинформационных технологий, предоставляемая конечным пользователям в качестве услуги. Примерами услуг ИТ могут служить передача сообщений, бизнес-приложения, сервисы файлов и печати, сетевые сервисы и т. д.»[9]

Управление сервис-ориентированной ИС[10] осуществляется с учетом методологий ITIL V3, Cobit, стандарта ГОСТ Р ИСО/МЭК 20000[11]. Процессы управления в ITIL подразделяются на процессы поддержки и предоставления ИТ-услуг (рис. 5). В соответствии с ITIL все процессы должны измеряться и иметь наборы метрик.

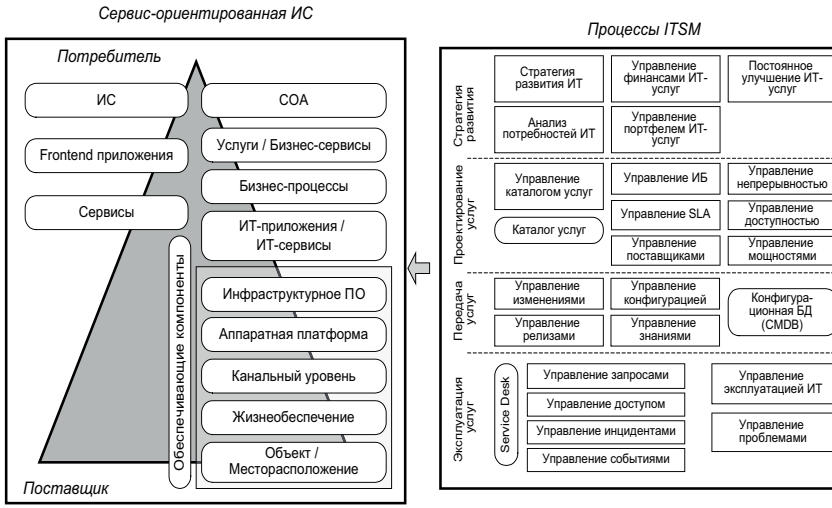


Рис. 5. Структура процессов управления сервис-ориентированной ИС

В сервис-ориентированной ИС подлежат измерению не только процессы, но и сервисы. При создании метрик сервисов необходимо учитывать следующие аспекты:

– в случае, если ИТ-сервисы непосредственно участвуют в реализации бизнес-сервисов, их показатели напрямую связаны с показателями бизнес-сервисов и значения метрик ИТ-сервисов формируются автоматически информационной системой по результатам бизнес-деятельности;

– поскольку ИТ-сервисы могут предоставляться в виде ИТ-услуг потребителям, необходимо проводить оценку уровня предоставления ИТ-сервисов в соответствии с заключенным между поставщиком и потребителем соглашением об уровне предоставления сервисов (SLA); в этом случае применяются метрики отклонений от KPI, позволяющие проанализировать способность ИТ-подразделений к обеспечению требуемого уровня ИТ-сервисов, зафиксированных в SLA;

– для обеспечения требуемого уровня показателей как отдельных ИТ-сервисов, так и всей ИС в целом необходима оценка процессов, реализуемых в ИТ-подразделениях.

В соответствии с ITIL V3[12] к формированию метрик ИТ-сервисов применяется подход, основанный на иерархии метрик:

– первый уровень – метрики сервисов и метрики составляющих их компонентов;

– второй уровень – суммарные метрики сервисов и метрики процессов управления;

– третий уровень – метрики ИТ-целей;

– четвертый уровень – метрики бизнес-целей.

В основе данного подхода лежит многоуровневая архитектура ИС и принцип взаимосвязи бизнес-сервисов и ИТ-сервисов.

Если ИТ-сервисы рассматриваются в виде услуг, предоставляемых пользователям, то с позиций ITSM все метрики, используемые для описания и оценки ИТ-сервисов, подразделяются на три группы:

– сервисные (показывают уровень предоставления сервиса);

– технологические (показывают характеристики инфраструктуры);

– процессные (показывают эффективность внутренних процессов ИТ в организации).

Таким образом, для сервис-ориентированной ИС при разработке метрик необходимо учитывать подход на основе иерархии метрик и подход сервис-менеджмента к формированию метрик для оценки ИТ-услуг.

Формирование системы метрик при реализации проекта ИС

Анализ методологий RUP, ITIL, BPM, проектного менеджмента показал необходимость проведения измерений на всех этапах проекта и жизненного цикла сервис-ориентированной информационной системы. Измерения являются важной составляющей систем управления проектом и ЖЦ ИС и направлены прежде всего на измерение процессов и сервисов.

Осуществление измерений требует системного подхода, который позволит объединить различные меры и метрики с целью формирования единого взгляда на процессы и сервисы в контексте их измерения[13].

В соответствии с методологией ITIL измерение как процесс также состоит из последовательности этапов и шагов.

Этап 1. Подготовка проведения измерений: определение целей измерений; определение, что необходимо измерять, чтобы получить информацию, необходимую для принятия решений; определение, что можно измерить; определение, какими средствами может быть получена требуемая информация и данные; формирование

целевых показателей; выбор средств измерений; выбор методов осуществления мониторинга и измерений; определение ответственных за управление данными; определение, кем и с помощью каких методов и критериев будут обрабатываться и анализироваться данные; определение видов отчетов.

Этап 2. Сбор данных.

Этап 3. Обработка данных измерений.

Этап 4. Анализ данных.

Этап 5. Представление информации.

Этап 6. Разработка корректирующих действий.

На этапе подготовки проведения измерений одной из составляющих является разработка системы показателей. Наиболее часто в качестве показателей применяются индикаторы или метрики – доступные наблюдению и измерению характеристики процессов, сервисов и их взаимосвязей, по значению которых можно судить о состоянии, изменениях и поведении создаваемой или эксплуатируемой системы.

При разработке метрик учитывают ряд принципов.

1. SMART (Specific, Measurable, Achievable, Realistic, Timely – конкретный, измеримый, достижимый, реалистичный, своевременный) – набор характеристик любой метрики, которые должны быть определены, прежде чем она будет применена.

Метрика должна относиться к конкретному процессу (или части процесса), сервису. Метрика должна быть измерима и достижима при имеющихся условиях. Метрика должна быть реалистична, иметь смысл и измерять реальные процессы и сервисы. Метрика должна быть своевременной, иметь обоснованную частоту измерений и срок применения.

2. KISS (Keep It Simple Stupid) – метрика должна быть хорошо разъяснена и способы ее достижения достаточно понятны.

3. GQM (Goal, Question, Metric – цель, вопрос, метрика) – подход к формированию метрик сверху вниз – от определения высокоуровневых целей (проекта, процесса, сервиса) к составлению перечня вопросов, позволяющих определить степень достижения цели, а затем разработка набора метрик для количественной оценки результатов.

4. MAPE (Mean Absolute Percentage Error – средняя абсолютная ошибка в процентах) – подход к обеспечению сравнимости метрик. Для упрощения понимания абсолютных численных показателей и сопоставления метрик используется процентное представление числовых величин. Процентные величины применяются также для оценки достоверности прогнозируемых метрик.

Таким образом, учитывая рекомендации выше рассмотренных методологий, систему метрик сервис-ориентированной ИС целесообразно формировать с учетом уровней управления и типов целей (рис. 6).

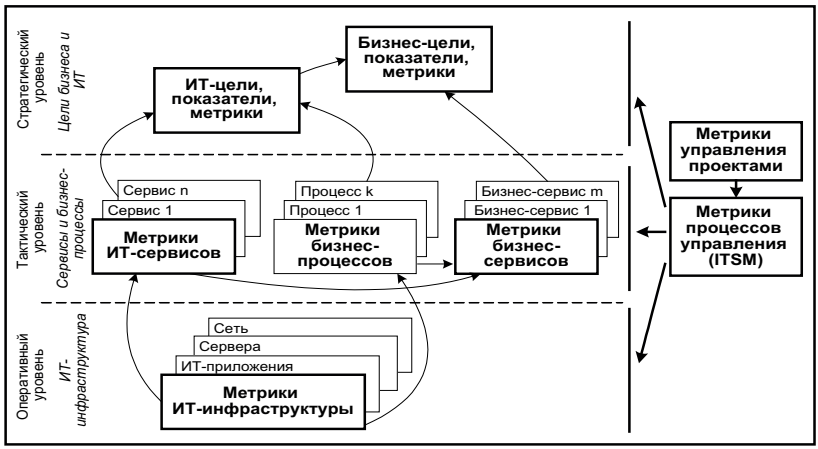


Рис. 6. Структура системы метрик сервис-ориентированной ИС

Метрики процессов управления группируются с учетом типов целей: краткосрочным соответствуют операционные процессы, среднесрочным – тактические, долгосрочным – стратегические.

Для операционных процессов управления используются метрики: службы Service Desk; управления инцидентами; управления конфигурацией; управления изменениями; управления релизами; для поддержки приложений; для разработки приложений; управления инфраструктурой ИКТ.

Для тактических процессов управления применяются метрики: управления уровнем предоставления сервисов (SLA); управления проблемами; управления мощностями; управления непрерывностью предоставления ИТ-сервисов; управления доступностью; управления информационной безопасностью.

Для стратегических процессов управления применяются метрики: постоянного улучшения ИТ-сервисов и услуг; управления финансами для ИТ-услуг; управления рисками; управления документацией; управления компетентностью и обучением сотрудников.

Метрики для управления проектами включают: число работ, не выполненных в запланированные сроки; число выявленных рисков

проекта; общее время задержки проекта; задержка критического пути; удовлетворенность клиентов и др.

Состав метрик ИТ-инфраструктуры определяется перечнем характеристик, необходимых для формирования базы данных управления конфигурациями (CMDB). На метрики ИТ-сервисов влияют характеристики и показатели, используемые для ведения каталога сервисов.

Таким образом, создание системы метрик – одна из важнейших задач при реализации проекта ИС.

Заключение

Проведенный анализ показал, что важной составляющей мониторинга и контроля является формирование системы показателей и метрик, разработанных с учетом определенных методологий, методов и стандартов, и только на основе таких метрик может осуществляться анализ и оценка ИТ-сервисов и процессов.

В соответствии с методологией процессного подхода одной из основных характеристик процесса является его измеримость, т. е. каждый процесс должен иметь систему показателей и метрик. В методологии ITIL рекомендуется формировать метрики одновременно с процессами. Необходимость разработки системы показателей и метрик процессов еще на этапе моделирования бизнес-процессов отмечается в методологии RUP.

Кроме того, постоянный рост интереса к бизнес-аналитике, в особенности к процессно-ориентированной, необходимость интеграции существующих и разрабатываемых информационных систем с BI-системами, требует продуманного, научно обоснованного подхода к формированию показателей и метрик сервисов и процессов.

Примечания

[1] См.: *Федоров И.Г.* О терминологии процессного управления // Открытое образование. 2013. № 4 (99). С. 32–39.

[2] См.: ГОСТ Р ИСО 9000-2008. «Системы менеджмента качества. Основные положения и словарь» («Quality management systems. Fundamentals and vocabulary»). М.: Стандартинформ, 2008. 36 с.

[3] См.: Там же.

[4] См.: Там же.

[5] См.: Там же; Глоссарий терминов и определений (Glossary Terms and Definitions) [Электронный ресурс] // IT Expert. URL: http://www.itexpert.ru/

rus/biblio/itil_v3/ITILV3_Glossary_Russian_v092_2009.pdf (дата обращения: 30.04.2014); ГОСТ Р ИСО/МЭК 20000-1-2010. «Информационная технология. Менеджмент услуг». Ч. 1. Спецификация. М.: Стандартинформ, 2011. IV, 15 с.

[6] См.: Глоссарий терминов и определений [Электронный ресурс].

[7] См.: ГОСТ Р ИСО/МЭК 15288-2005. «Информационная технология. Системная инженерия. Процессы жизненного цикла систем». М.: Стандартинформ, 2005. 57 с.

[8] См.: *Кролл П., Кратчен Ф.* Rational Unified Process – это легко. Руководство по RUP / Пер. с англ. М.: КУДИЦ-ОБРАЗ, 2004. 432 с.

[9] См.: ГОСТ Р 53114-2008. «Защита информации. Обеспечение информационной безопасности в организации. Основные термины и определения». М.: Стандартинформ, 2009. IV, 15 с.

[10] См.: *Сатунина А.Е., Сысоева Л.А.* Анализ моделей управления сервис-ориентированной информационной системой // Вестник РГГУ. Сер. «Информатика. Защита информации. Математика». 2013. № 14 (115). С. 182–193.

[11] См.: ГОСТ Р ИСО/МЭК 20000-1-2010; ГОСТ Р ИСО/МЭК 20000-2-2010. «Информационная технология. Менеджмент услуг». Ч. 2. Кодекс практической деятельности. М.: Стандартинформ, 2011. VI, 28 с.; ITIL [Электронный ресурс]. URL: http://www.itil.co.uk (дата обращения: 30.04.2014); COBIT 5: A Business Framework for the Governance and Management of Enterprise [Электронный ресурс] // ISACA. URL: http://www.isaca.org/cobit (дата обращения: 30.04.2014).

[12] См.: ITIL [Электронный ресурс].

[13] См.: Глоссарий терминов и определений [Электронный ресурс].

А.В. Лаврентьев

ОПТИМИЗАЦИЯ МНОГОКОМПОНЕНТНЫХ ПРИЛОЖЕНИЙ В СРЕДЕ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ С НЕСКОЛЬКИМИ ПРОВАЙДЕРАМИ

В статье рассматривается подход гибридной двухшаговой оптимизации размещения многокомпонентных приложений в среде, состоящей из множества облаков. Предложен основанный на проблеме частично целочисленного линейного программирования метод автоматизированного выбора облачных провайдеров, распределения нагрузки между ними, а также оптимизации облачных архитектур с целью минимизации стоимости использования облачных ресурсов при условии сохранения гарантированного уровня качества обслуживания.

Ключевые слова: мультиоблако, многокомпонентное приложение, теория массового обслуживания, линейное программирование, локальный поиск.

Развитие облачной парадигмы оказывает значительное влияние на информационно-телекоммуникационные технологии в последние годы. Облачные провайдеры предоставляют широкий спектр технологий и сервисов, беря на себя задачи управления и обслуживания инфраструктуры. Однако помимо очевидных преимуществ облачные вычисления привели к появлению новых проблем и важных вопросов в разработке приложений. В действительности текущие облачные технологии и их ценовые модели могут быть настолько комплексными и отличными друг от друга, что поиск дешевых решений, гарантирующих требуемое качество обслуживания, является очень трудоемкой задачей. Разработчик приложений, сталкивающийся с подобным вопросом,

должен рассматривать огромное множество облачных альтернатив и быть в состоянии оценить затраты и качество обслуживания для каждого из них. Кроме того, в то время как информация об архитектурах и ценах находится в открытом доступе, оценка качества обслуживания является гораздо более сложным и трудоемким процессом, потому что облачная среда часто арендуется множеством заказчиков и качество обслуживания изменяется с течением времени согласно текущему уровню загруженности и соревнованию за ресурсы между поддерживаемыми приложениями. Хотя существуют некоторые аналитические модели приблизительной оценки качества работы программных систем, до сих пор не предпринималось попыток принятия во внимание специфики облачной среды.

Конечно, имеются классические модели, разработанные с целью предсказания поведения статичных систем и ориентированные на постоянные во времени параметры, такие как уровень рабочей нагрузки, конфигурацию оборудования и показатели качества. Например, Palladio Component Model[1] (далее – PCM) и Palladio Bench. PCM – предметно-ориентированный язык для описания архитектуры приложений и ресурсов и анализа нефункциональных требований, но он ориентирован на промышленные системы, оценка качества может быть проведена только для пиковых нагрузок, и у него ограниченная поддержка облачных архитектур. В противоположность этому облачно-ориентированные системы динамичны. В них для оценки качества и стоимости требуется учитывать параметры, крайне зависимые от времени (эластичность облачных архитектур подразумевает значительные суточные колебания ресурсов, поддерживающих выполняемые задачи). Следует иметь в виду, что помимо анализа цен и качества, присутствует проблема быстрого и эффективного анализа пространства возможных облачных конфигураций в автоматическом или полуавтоматическом режиме.

В данной работе предлагается новый метод быстрого поиска облачных конфигураций минимальной стоимости для заданного приложения на основе проблемы частично целочисленного линейного программирования. Допустимость конечных решений задана с помощью нефункциональных условий модели.

Кроме того, решения предложенной проблемы линейного программирования дополнительно оптимизируются с использованием алгоритма, основанного на локальном поиске, который также спроектирован с целью получения и анализа приемлемых альтернатив среди множества облачных конфигураций.

Расширение языка Palladio Component Model для описания среды облачных вычислений

Метамодели реальных облачных приложений, используемые как вводная информация для проблемы линейного программирования, составляются с помощью расширения языка Palladio Component Model для облачных приложений. Это расширение позволяет представить различные условия и зависимые от времени профили для параметров, используемых при оценке качества. Чистый PCM позволяет проектировать различные аспекты приложений через построение специализированных диаграмм.

Для примера рассмотрим приложение Apache Open For Business (далее – OfBiz). Рис. 1 и 2 показывают всю необходимую информацию об этом приложении с помощью UML-подобного описания.

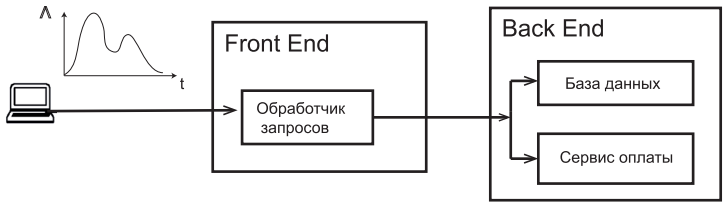


Рис. 1. Структура приложения OfBiz

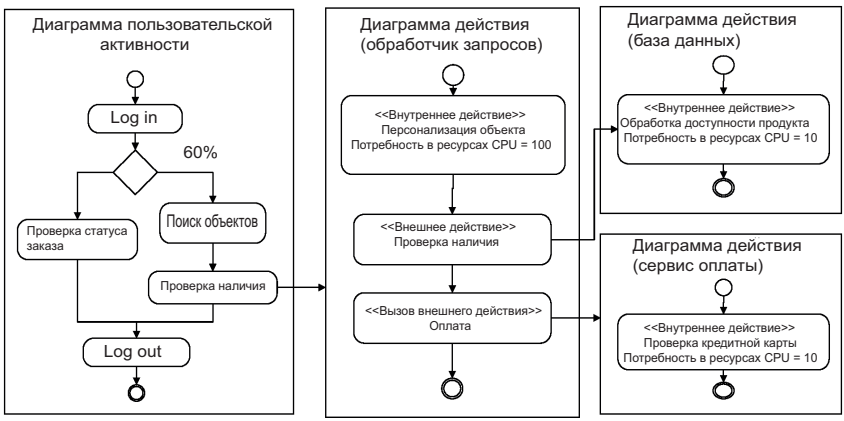


Рис. 2. Примеры диаграмм действия для приложения OfBiz

OfBiz – промышленное приложение с открытым исходным кодом, разработанное Apache Software Foundation и используемое множеством компаний. Оно обладает широкой функциональностью, требуемой для таких систем, как ERP, CRM, SCM или электронная коммерция. Приложение для электронной коммерции является хорошим кандидатом для реализации в облачной среде из-за высокой интерактивности с потенциально большим количеством пользователей.

Левая диаграмма рис. 2 показывает обычное поведение пользователей при работе с этим приложением. В данном примере 60 % пользователей используют приложение для оформления покупки некоторого продукта, в то время как оставшиеся 40 % проверяют статус доставки продукта, заказанного ранее. Входящая рабочая нагрузка представлена в виде количества запросов в секунду. Расширение для языка PCM позволяет задать рабочую нагрузку отдельно для каждого из 24 часов. Все запросы, сформированные пользователями, обрабатываются компонентом <Обработчик запросов>. Поведение функционала <Проверка наличия> описывается диаграммой деятельности, соответствующей обработчику. Для выполнения запроса Front End виртуальной машине требуются внутренние вычисления (например, для получения цены продукта), влияние которых на физические ресурсы хоста обозначено <Потребность в ресурсах CPU>, и взаимодействие с некоторыми компонентами, расположенными на Back End. В нашем примере запрос обращается к компоненту <База данных> для проверки доступности выбранного продукта и к компоненту <Сервис оплаты> для проверки корректности информации о кредитной карте, введенной пользователем. Рис. 1 демонстрирует размещение компонентов между виртуальными машинами.

Стандартный PCM позволяет разработчикам приложений создавать диаграммы на основе подобной информации и получать из них модель многоуровневой сети очередей (Layered Queuing Network[2] (далее – LQN)). Оценка качества может быть получена из модели LQN аналитически или посредством имитации с помощью LQNS[3] (Layered Queuing Network Solver) – типового инструмента для решения таких моделей. Предполагаем, что размещение компонентов на вычислительных уровнях (например, Back End и Front End) уже было осуществлено разработчиком и не будет меняться в течение оптимизационного процесса. LQN модели позволяют получить множество оценок качества; в этой работе будут использованы время ответа и стоимость.

Двухшаговый гибридный алгоритм оптимизации

В среде облачных вычислений стоимость инфраструктуры сложно вычислить, так как ценовые политики, предоставляемые различными облачными провайдерами, очень гетерогенны. В этой работе под стоимостью понимается сумма цен арендованных ресурсов, оформленных на условиях оплаты по факту использования. В этом случае виртуальная машина арендуется на час, по истечении которого ее использование может быть снова продлено и оплачено, либо она будет отключена. Этот тип ценовой политики наиболее общий для всех облачных провайдеров. Основная цель подобного ценового моделирования заключается в том, чтобы показать, как вопросы цены облачных вычислений могут быть включены в процесс оптимизации.

Как основную оценку качества будем использовать время ответа на запросы пользователей. Также предполагается, что вычислительный сегмент образован на основе локальной сети (например, все виртуальные машины этого сегмента расположены в той же самой зоне доступности в Amazon EC2), так что взаимодействие между различными уровнями приложения не приводит к превышению доступной пропускной способности сети.

В данной работе рассматривается метод двухшаговой гибридной оптимизации, позволяющий решить проблему распределения ресурсов в облачной среде. Первый шаг состоит в решении проблемы частично-целочисленного линейного программирования, в которой качество обслуживания развертываемого решения вычисляется посредством M/G/1-PS модели очередей. Такая модель позволяет вычислить среднее время ответа на запрос в закрытой форме. Цель данного шага в том, чтобы быстро определить аппроксимированное начальное решение, которое в дальнейшем будет улучшено посредством алгоритма локального поиска на шаге 2. Цель шага 2 в том, чтобы итеративно улучшать начальную конфигурацию облачной среды, рассматривая различные альтернативы. Алгоритм локального поиска основан на модели LQN, которая позволяет провести более точную оценку параметров качества обслуживания. Рис. 3 и 4 показывают поток работ в процессе оптимизации для шага 1 и шага 2 соответственно. Как было сказано ранее, приложение описывается диаграммами на языке PCM с соответствующим расширением. Информация, хранящаяся в этих диаграммах, используется как входная информация оптимизационной модели, которая является основной целью данной работы. С целью дальнейшего улучшения результаты оптимизации затем обрабатываются алгоритмом локального поиска на основе модели LQN.

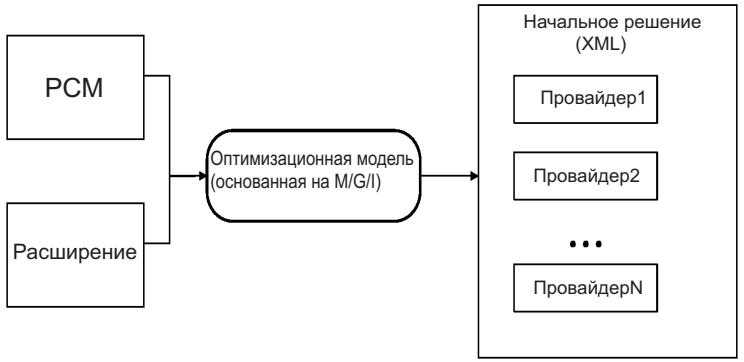


Рис. 3. Рабочий поток шага 1 оптимизации на основе модели M/G/1

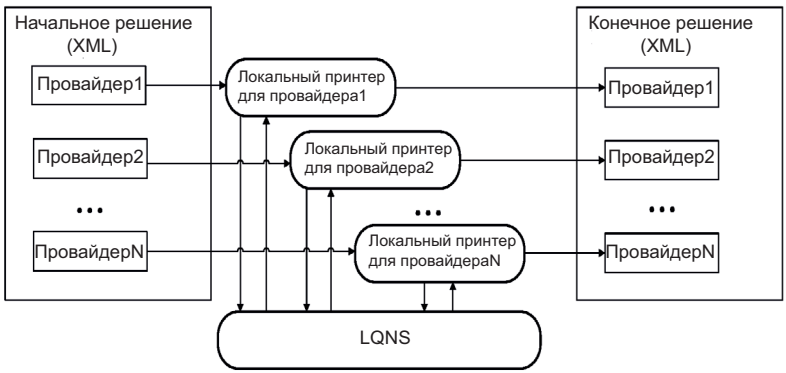


Рис. 4. Рабочий поток шага 2 оптимизации с помощью локального поиска

Цель оптимального выбора и распределения облачных ресурсов заключается в минимизации стоимости использования при одновременном удовлетворении некоторых заданных пользователем ограничений. Как было сказано ранее, приложение может быть представлено как композиция нескольких компонентов *C*, каждый из которых соответствует некоторому множеству функционалов *K* с заданной потребностью в ресурсах. Каждый компонент размещен в некотором пуле ресурсов (уровне архитектуры) *I*, сформированном множеством гомогенных виртуальных машин. Такое множество не статично, но может масштабироваться в соответствии с изменением входящей рабочей нагрузки. Так как дневная рабо-

чая нагрузка периодична для многих приложений[4], дальнейший анализ будет проводиться с учетом временного горизонта в один день. Многие облачные провайдеры взимают почасовую оплату за пользование ресурсами, так что будет разумно разбить временной горизонт на 24 части по одному часу каждая. Для простоты в дальнейшем будем полагать, что ограничения на время ответа для оптимизируемого приложения заданы заранее.

Рис. 5 демонстрирует модель иерархического распределения нагрузок в мультиоблачной[5] среде, которая состоит из нескольких отдельных облаков, предоставляемых различными провайдерами. В каждом облаке размещены идентичные копии приложения. Пользовательский запрос поступает на первый внешний балансировщик нагрузок, который определяет, какому провайдеру P будет адресован конкретный запрос (здесь и далее в работе под облачным провайдером понимается именно IaaS провайдер). Затем балансировщик нагрузок выбранного провайдера определяет сегмент облачной системы, который будет обрабатывать запросы. Сегмент, с точки зрения рассматриваемой задачи, может быть представлен в виде уровней (например, Front End и Back End, как в примере OfBiz). На каждом уровне запросы обрабатываются некоторым количеством виртуальных машин. Общая рабочая нагрузка обозначается Λ_t (запросов в секунду).

Пользователи взаимодействуют с приложением, формируя запросы. Множество возможных запросов обозначим K. Более того, каждый класс запросов характеризуется вероятностью его выполнения α_k и множеством компонентов, его поддерживающих (т. е. расположенных на пути выполнения). Наконец, будем предполагать, что запросы обрабатываются согласно политике PS[6] (processor sharing), типичной политике планирования в веб-приложениях, которая равномерно распределяет нагрузку между всеми виртуальными машинами в пуле. Требования к качеству обслуживания в модели представлены задаваемым разработчиком приложения ограничением на среднее время ответа $\{R\}_k$ для множества классов K.

Проблема, которой посвящена данная работа, может быть представлена в виде четырех аспектов принятия решений: первый заключается в выборе некоторого подмножества провайдеров, на виртуальных машинах которых будет развернуто приложение; второй задает распределение входящей рабочей нагрузки между выбранными провайдерами; третий описывает выбор типа виртуальных машин V для каждого пула ресурсов; в то время как четвертый определяет масштабирование множества виртуальных машин

в каждом пуле с целью выполнения заданных пользователем ограничений, т. е. вычисление оптимального количества виртуальных машин, которые требуется выделить в каждом пуле в каждый временной промежуток.

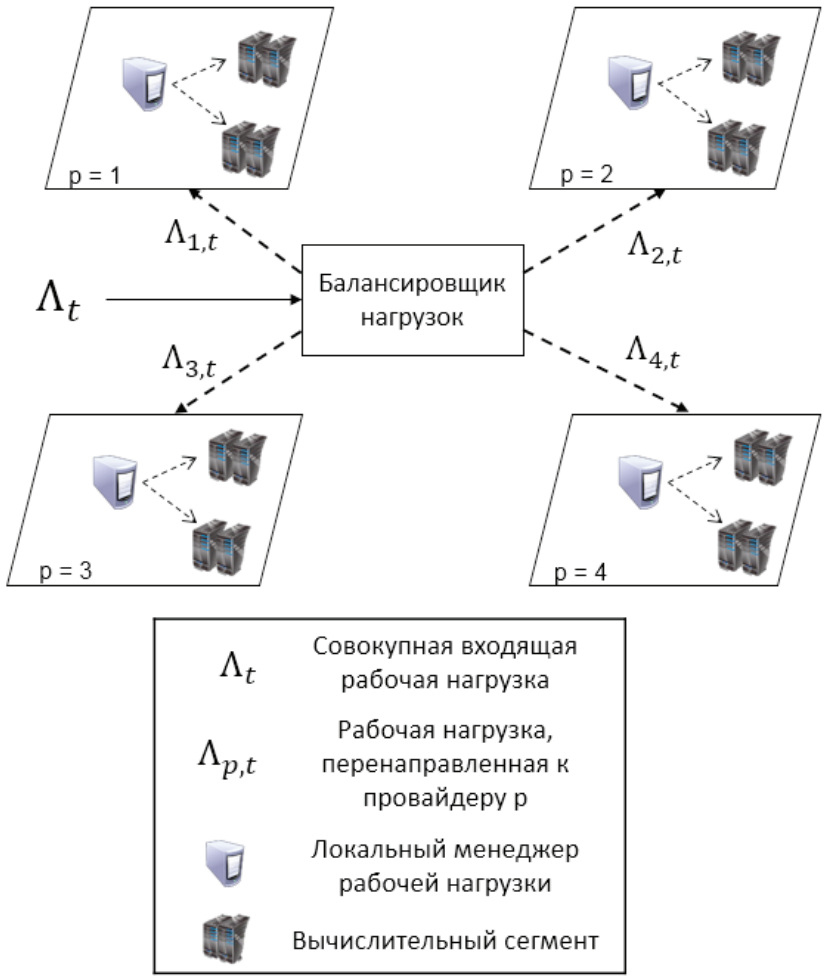


Рис. 5. Модель иерархического распределения рабочей нагрузки

Модель оптимизации на основе M/G/1

В свете высказанных соображений задача оптимального распределения ресурсов может быть сформулирована как:

$$\min_{z,v} \sum_{i\in I} \sum_{p\in P} \sum_{v\in V_p} \sum_{t\in T} C_{p,v,t} Z_{p,i,v,t} \tag{1}$$

При условии:

$$\sum_{p\in P} x_p \geq F, \forall p \in P \ , \tag{2}$$

$$\sum_{v\in V_p} w_{p,i,v} = x_p, \forall p \in P, \forall i \in I \ , \tag{3}$$

$$w_{p,i,v} \leq z_{p,i,v,t}, \forall p \in P, \forall i \in I, \forall v \in V_p, \forall t \in T \ , \tag{4}$$

$$z_{p,i,v,t} \leq N w_{p,i,v}, \forall p \in P, \forall i \in I, \forall v \in V_p, \forall t \in T \ , \tag{5}$$

$$\sum_{v\in V_p} M_{p,v} w_{p,i,v} \geq \{M\}_{p,i}, \forall p \in P, \forall i \in I \ , \tag{6}$$

$$\sum_{v\in V_p} S_{p,v} z_{p,i_c,v,t} > G_{p,k,c,t}, \forall p \in P, \forall k \in K, \forall c \in C, \forall t \in T, \tag{7}$$

$$\sum_{v\in V_p} (1 - \mu_{k,c}\{R\}_{k,c} S_{p,v}) z_{p,i_c,v,t} \leq \mu_{k,c} G_{p,k,c,t} \{R\}_{k,c}, \quad \forall p \in P, \forall k \in K, \forall c \in C, \forall t \in T \ , \tag{8}$$

$$\sum_{p\in P} \wedge_{p,t} = \wedge_t, \forall t \in T \ , \tag{9}$$

$$x_p \gamma_p \wedge_t \leq \wedge_{p,t}, \forall p \in P, \forall t \in T \ , \tag{10}$$

$$\wedge_{p,t} \leq x_p \wedge_t, \forall p \in P, \forall t \in T \ , \tag{11}$$

$$z_{p,i,v,t} \in \mathrm{N}_o, \forall p \in P, \forall i \in I, \forall v \in V_p, \forall t \in T \ , \tag{12}$$

$$w_{p,i,v} \in \{0,1\}, \forall p \in P, \forall i \in I, \forall v \in V_p \ , \tag{13}$$

$$x_p \in \{0,1\}, \forall p \in P \ , \tag{14}$$

$$G_{p,k,c,t} = \wedge_{p,t} \sum_{\tilde{c}\in I_c} \sum_{\tilde{k}\in K} \frac{\alpha_k \rho_{k,\tilde{c}}}{\mu_{\tilde{k},\tilde{c}}} \ . \tag{15}$$

Табл. 1 содержит множество индексов оптимизационной модели, табл. 2 – список параметров и табл. 3 описывает переменные решения.

Таблица 1

Множества индексов

$t \in T$	Временной интервал
$i \in I$	Пул ресурсов
$p \in P$	Провайдер
$v \in V_p$	Тип виртуальных машин провайдера p
$k \in K$	Класс запросов
$c \in C$	Компонент

Таблица 2

Параметры модели

Λ_t	Входящая рабочая нагрузка в момент t
α_k	Процент запросов класса k в рабочей нагрузке
γ_p	Минимально допустимый процент запросов, перенаправляемых провайдеру p
$p_{k,c}$	Вероятность обработки компонентом c класса запросов k
$\mu_{p,v,k,c}$	Максимальная интенсивность обработки запросов класса k компонентом c, поддерживаемым виртуальными машинами типа v провайдера p
U_k	Множество компонентов, обрабатывающих запросы класса k
I_c	Множество компонентов, размещенных на том же пуле ресурсов, что и c
$C_{p,v,t}$	Цена единственной виртуальной машины типа v провайдера p в момент времени t
$M_{p,v}$	Память виртуальных машин типа v провайдера p
$\{M\}_{p,i}$	Ограничение памяти для пула ресурсов i провайдера p
$\{R\}_{k,c}$	Максимальное допустимое среднее время ответа для запросов класса k в компоненте c
F	Минимально допустимая мощность множества выбранных провайдеров

Таблица 3

Целевые переменные

x_p	Двоичная переменная, равная 1, если выбран провайдер p, и 0 в противном случае
$\Lambda_{p,t}$	Количество запросов в секунду, перенаправляемых провайдеру p во временном интервале t
$w_{p,i,v}$	Двоичная переменная, равная 1, если тип виртуальных машин v используется в пуле ресурсов i провайдера p, и 0 в противном случае
$z_{p,i,v,t}$	Количество виртуальных машин типа v в пуле ресурсов i в момент времени t

Выражение (1) представляет итоговую дневную стоимость использования облачных ресурсов, которые требуется минимизировать.

Для каждого пула ресурсов:

– условия (13) и (3) гарантируют, что будет выбран только один тип виртуальных машин;

– условие (12) задает целочисленность количества виртуальных машин;

– условие (4) гарантирует не пустое множество виртуальных машин, а (5) их гомогенность;

– условия (2) и (14) определяют, что будут выбраны хотя бы F облачных провайдеров;

– условия (9), (10) и (11) описывают распределение входящей рабочей нагрузки между выбранными провайдерами так, чтобы была распределена вся рабочая нагрузка (условие (9)), каждому выбранному провайдеру был передан ее процент (условие (10)) и никакому провайдеру не был присвоен объем входящей нагрузки, превышающий реально имеющуюся (условие (11)).

Наконец, (6) и (8) представляют ограничения на память и время ответа соответственно, в то время как (7) является условием равновесия для модели M/G/1.

Как было обсуждено ранее, для оценки среднего времени ответа облачного приложения мы моделируем каждый пул ресурсов как очередь M/G/1. Так или иначе, в общем запрос класса k обрабатывается более чем одним компонентом. Пусть $\Lambda_{p,k,t} = \alpha_k \Lambda_{p,t}$ – входящая рабочая нагрузка во время t для класса запроса k, перенаправляемая провайдеру p, и $\Lambda_{p,k,c,t} = p_{k,c} \Lambda_{p,k,t}$ – интенсивность входного потока запросов класса k в компонент c. Время ответа для запросов класса k может быть получено как:

$$R_{k,t} = \sum_{c \in U_k} \rho_{k,c} R_{k,c,t} = \sum_{c \in U_k} \rho_{k,c} \frac{\dfrac{1}{\mu_{p,\tilde{v},\tilde{k},\tilde{c}}}}{\sum_{\tilde{c} \in I_c} \sum_{\tilde{k} \in K} \dfrac{\wedge_{p,\tilde{k},\tilde{c},t}}{\mu_{p,\tilde{v},\tilde{k},\tilde{c}} z_{p,i_c,\tilde{v},t}}}, \qquad (16)$$

где U_k – множество компонентов, обрабатывающих запрос k, I_c отражает множество компонентов, которые размещены на той же виртуальной машине, что и c. Другими словами, среднее время ответа получено суммированием времени, потраченного запросом на каждый компонент, умноженным на вероятность того, что запрос действительно будет обработан компонентом. Заметим, что время ответа для компонента c во время t зависит от типа и количества виртуальных машин в этот момент в пуле, в котором расположен этот компонент.

С целью упрощения уравнения (16) рассмотрим тип виртуальных машин с наименьшей скоростью CPU и, приняв его за опорный, вычислим интенсивность обслуживания для каждого класса и компонента $\mu_{k,c}$. Тогда интенсивность обслуживания для остальных типов виртуальных машин может быть выражена через интенсивность обслуживания опорного как:

$$\mu_{p,v,k,c} = \mu_{k,c} S_{p,i_c} = \mu_{k,c} \sum_{v \in V_p} S_{p,v} w_{p,i_c,v}, \qquad (17)$$

где $S_{p,v}$ – отношение скоростей между опорным типом и типом v провайдера p, в то время как i_c – индекс пула ресурсов, в котором размещен компонент c.

Пусть $z_{p,i,t}$ – количество виртуальных машин некоторого типа (только один тип может быть выбран согласно (3)) для пула ресурсов i во время t. Тогда выполняются следующие соотношения:

$$z_{p,i,t} = \sum_{v \in V_p} z_{p,i,v,t}, \qquad (18)$$

$$\mu_{p,v,k,c} z_{p,i_c,v,t} = \mu_{k,c} S_{p,i_c} z_{p,i_c,t} = \mu_{k,c} \sum_{v \in V_p} S_{p,v} z_{p,i_c,v,t}. \qquad (19)$$

Время ответа для запроса k, обрабатываемого компонентом c, поддерживаемым виртуальными машинами типа v, из предположения M/G/1 может быть вычислено как:

$$R_{k,c,t} = \frac{\dfrac{1}{\mu_{k,c} S_{p,i_c}}}{1 - \sum_{\tilde{c} \in I_c} \sum_{\tilde{k} \in K} \dfrac{\wedge_{p,\tilde{k},\tilde{c},t}}{\mu_{\tilde{k},\tilde{c}} S_{p,i_c} z_{p,i_c,t}}} \quad . \qquad (20)$$

Через замену (17) и (19) в (16) мы можем записать следующее ограничение на время ответа для класса k:

$$R_{k,t} = \sum_{c \in U_k} \rho_{k,c} R_{k,c,t} = \sum_{c \in U_k} \rho_{k,c} \frac{\dfrac{1}{\mu_{k,c} S_{p,i_c}}}{1 - \dfrac{\wedge_{p,t}}{S_{p,i_c} z_{p,i_c,t}} \sum_{\tilde{c} \in I_c} \sum_{\tilde{\kappa} \in K} \dfrac{\alpha_{\tilde{\kappa}} \rho_{\tilde{\kappa},\tilde{c}}}{\mu_{\tilde{\kappa},\tilde{c}}}} \le \{R\}_k \ . \quad (21)$$

Уравнение (21) нелинейно из-за присутствия $S_{p,t} z_{p,i,t}$ в знаменателе. С целью получения линейной модели, которая могла бы быть эффективно решена с помощью соответствующего программного обеспечения (например CPLEX), разобьем его на множество более строгих условий, задающих ограничения на время ответа для каждого компонента, обрабатывающего запрос.

Для этого распределим границу времени ответа между компонентами пути и используем наиболее строгое ограничение среди всех ограничений, сгенерированных по всем возможным путям, по которым может пройти запрос. Другими словами, пусть:

$$r_{k,c,u} = \begin{cases} \dfrac{\dfrac{1}{\mu_{k,c}}}{\sum_{\partial \varepsilon u} \dfrac{1}{\mu_{k,c}}}, & \text{если } c \text{ принадлежит пути, или} \\ 0, & \text{если иначе.} \end{cases} \qquad (22)$$

И пусть:

$$\{R\}_{k,c} = \min_u r_{k,c,u} \{R\}_k \ . \qquad (23)$$

Вместо использования ограничения (20) для времени ответа введем семейство ограничений:

$$\{R\}_{k,c,t} = \{R\}_{k,c} \qquad (24)$$

и после нескольких простых алгебраических преобразований получим ограничение (8).

Наконец, условие (7) представляет собой условие равновесия для модели M/G/1, полученное из условия (20). Оно гарантирует, что знаменатель в (20) будет больше нуля.

Алгоритм локального поиска

Теперь опишем алгоритм локального поиска, используемый на следующем шаге гибридной метаэвристической оптимизации для повышения качества полученных результатов. Основным отличием между двумя оптимизационными процессами является то, что локальный поиск основан на модели LQN, являющейся более комплексной и точной, чем модель M/G/1, используемая в оптимизационной проблеме на предыдущем шаге. Другое отличие заключается в том, как алгоритм локального поиска просматривает пространство возможных альтернатив. Он использует эвристический подход, который разбивает проблему на два уровня, определяя типы виртуальных машин на верхнем уровне и их количество на нижнем.

Первый уровень реализует стохастический поиск с памятью о совершенных шагах. На каждой итерации тип виртуальных машин, используемый для некоторого пула, изменяется случайным выбором из всего множества доступных типов согласно архитектурным ограничениям. Запоминание шагов требуется для того, чтобы избежать циклического выбора кандидатов с той же самой конфигурацией. В тот момент, когда тип виртуальных машин изменен и зафиксирован, начинается процесс наращивания количества виртуальных машин до того момента, пока итоговая конфигурация не станет снова допустимой. (Изменение типа на более медленный может сделать текущую конфигурацию системы недопустимой.) Затем решение обрабатывается пошаговым уменьшением на 1 количества виртуальных машин в каждом пуле, пока оптимальное распределение ресурсов не будет найдено. Весь процесс повторяется заданное количество итераций, обновляя наилучшее решение каждый раз, как только допустимое и более дешевое решение будет найдено.

Заключение

В данной работе был предложен метод гибридной оптимизации облачных приложений. Минимизация стоимости использования облачных ресурсов с использованием нескольких облачных провайдеров осуществляется в два шага. На шаге 1 решается частично-целочисленная проблема линейного программирования, результаты которой предоставляют начальное решение для процедуры локального поиска (шага 2). Предложенный подход позволяет снизить стоимость облачных ресурсов и увеличить качество конечной системы, используя автоматизированный поиск альтернатив с лучшими характеристиками.

Дальнейшее исследование будет посвящено расширению подхода на PaaS системы. Другим направлением будущих разработок является включение метрик доступности в оптимизационную модель.

Примечания

[1] См.: *Becker S., Koziolek H., Reussner R.* The Palladio Component Model for Model-driven Performance Prediction // Journal of Systems and Software. 2009. Vol. 82 (1). P. 3–22.

[2] См.: *Neilson J.E., Woodside C.M., Petriu D.C., Majumdar S.* Software Bottlenecking in Client-server Systems and Rendezvous Networks // IEEE Transactions on Software Engineering. 1995. Vol. 21 (9). P. 776–782.

[3] *Franks G., Hubbard A., Majumdar S., Neilson J., Petriu D., Rolia J., Woodside M.* A Toolset for Performance Engineering and Software Design of Client-server Systems // Performance Evaluation. 1996. Vol. 24. P. 1–2.

[4] См.: *Birke R., Chen L.Y., Smirni E.* Data Centers in the Cloud: A Large Scale Performance Study // Proceedings of the 5th IEEE International Conference on Cloud Computing. Honolulu, 2012. P. 336–343.

[5] См.: *Celesti A., Tusa F., Villari M., Puliafito A.* How to Enhance Cloud Architectures to Enable Cross-Federation – Cloud Computing (CLOUD) // Proceedings of the 3rd IEEE International Conference on Cloud Computing. Miami, FL, 2010. P. 337–345.

[6] См.: *Kleinrock L.* Time-shared Systems: A Theoretical Treatment // Journal of the ACM. 1967. Vol. 14 (2). P. 242–261.

А.А. Пупыкина, А.Е. Сатунина

ФОРМАЛИЗАЦИЯ МЕТАМОДЕЛИ ВЕБ-ПРИЛОЖЕНИЯ

В статье рассматривается способ формализации метамодели веб-приложения для использования в процессе модельно-ориентированной разработки. Специфика предлагаемого подхода к проектированию веб-приложений заключается в разработке и использовании формализованного метода моделирования, основанного на использовании основных положений теории графов.

Ключевые слова: модель, модельно-ориентированный подход, трансформация, веб-приложение.

Объектом исследования является процесс модельно-ориентированной разработки веб-приложений, в частности, пользовательского интерфейса в Сети[1].

Детализация представления веб-приложения должна доходить до уровня, необходимого для генерации кода веб-приложения на конкретную платформу. Рациональным подходом будет являться постепенная детализация каждого аспекта моделирования по мере согласования модели верхнего уровня. Такой подход значительно снизит сложность процесса моделирования системы и дальнейшую поддержку моделей.

Методологическую основу предлагаемого подхода составляет метамодель веб-приложения, приведенная на рис. 1.

Трансформация платформенно-независимой модели в платформенно-зависимую осуществляется по заданной спецификации, которая содержит формальное однозначное описание преобразования модели общего вида к конкретной платформе реализации.

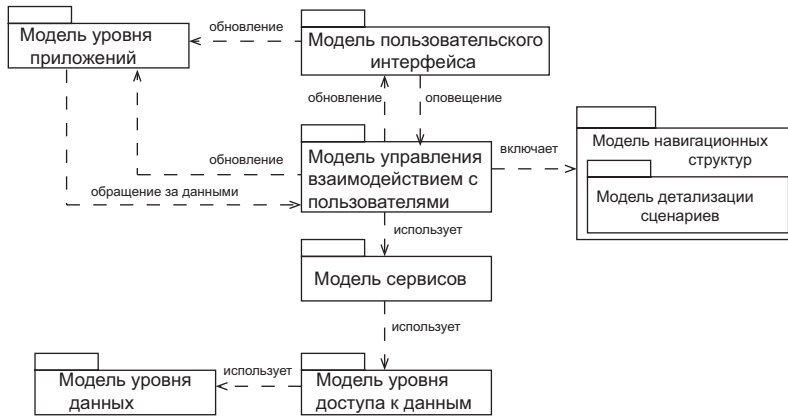


Рис. 1. Общая схема метамодели веб-приложения

Общая схема механизма трансформации представлена на рис. 2.

Рис. 2. Общая схема трансформации модели веб-приложения

Правила трансформации определяются для каждой платформы реализации. Поэтому актуальным становится вопрос добавления новых правил для расширения поддерживаемых технологических платформ инструментами модельно-ориентированной разработки пользовательских интерфейсов веб-приложений, а также модификации правил для совместимости с новыми версиями и предоставляемыми возможностями технологических платформ. Это приводит к необходимости применения автоматизированных процедур верификации и валидации механизма трансформации. Для реализации формулированной цели предлагается метамодели и правила трансформации представить в виде графов.

Метамодель веб-приложения

Представим метамодель в формализованном виде для определения состава, структуры и взаимосвязи элементов моделей.

Метамодель веб-приложения можно представить в виде множества:

$$M = (M_E, M_D, M_V, M_M, M_C, M_S)\text{, где}$$

M_E – модель уровня данных. $M_E = D_{CE}$, где D_{CE} – множество *UML*-диаграмм классов для моделирования сущностей предметной области:

$$D_{CE} = \left\{ d_{ND_{CE}}^{CE} \right\}, ND_{CE} = \overline{1, ND_{CE}},$$

ND_{CE} – количество диаграмм классов в модели данных.

M_D – модель уровня доступа к данным. $M_D = D_{CD}$, где D_{CD} – множество *UML*-диаграмм классов для моделирования объектов доступа к данным[2]:

$$D_{CD} = \left\{ d_{ND_{CD}}^{CD} \right\}, ND_{CD} = \overline{1, ND_{CD}},$$

ND_{CD} – количество диаграмм классов в модели уровня доступа к данным;

M_V – модель пользовательского интерфейса для моделирования наполнения веб-страниц визуальными элементами. $M_V = D_{CV}$, где D_{CV} – множество *UML*-диаграмм компонентов:

$$D_{CV} = \left\{ d_{ND_{CV}}^{CV} \right\}, ND_{CV} = \overline{1, ND_{CV}},$$

ND_{CV} – количество диаграмм компонентов в модели пользовательского интерфейса;

M_M – модель уровня модели интерфейса для представления предметной области наполнения веб-страниц визуальными элементами. $M_M = D_{CM}$, где D_{CM} – множество *UML*-диаграмм классов:

$$D_{CM} = \left\{ d_{ND_{CM}}^{CM} \right\}, ND_{CM} = \overline{1, ND_{CM}},$$

ND_{CM} – количество диаграмм классов в модели уровня модели интерфейса;

M_C – модель управления взаимодействием с пользователем. $M_C = (D_{UC}, D_{SM}, D_{CC})$, где D_{UC} – множество *UML*-диаграмм вариантов использования предназначенных для моделирования верхнего уровня навигационной архитектуры приложения и ролей доступа, D_{SM} – множество *UML*-диаграмм конечных автоматов для модели-

рования сценариев взаимодействия с пользователем, D_{CC} – множество *UML*-диаграмм классов для моделирования классов-контроллеров:

$$D_{UC} = \left\{ d^{UC}_{ND_{UC}} \right\}, ND_{UC} = \overline{1, ND_{UC}},$$

ND_{UC} – количество диаграмм вариантов использования в модели управления взаимодействием с пользователем;

$$D_{SM} = \left\{ d^{SM}_{ND_{SM}} \right\}, ND_{SM} = \overline{1, ND_{SM}},$$

ND_{SM} – количество диаграмм конечных автоматов в модели управления взаимодействием с пользователем;

$$D_{CC} = \left\{ d^{CC}_{ND_{CC}} \right\}, ND_{CC} = \overline{1, ND_{CC}},$$

ND_{CC} – количество диаграмм классов в модели управления взаимодействием с пользователем;

M_S – модель сервисов. $M_S = D_{CS}$, где D_{CS} – множество *UML*-диаграмм классов для моделирования сервисов:

$$D_{CS} = \left\{ d^{CS}_{ND_{CS}} \right\}, ND_{CS} = \overline{1, ND_{CS}},$$

ND_{CS} – количество диаграмм классов в модели сервисов.

Для описания моделей представим общую структуру (L,R)-помеченного T – типизированного графа диаграммы.

$$D_{ND_M} = \{ v^{ND_M}{}_j, e^{ND_M}{}_k, s^{ND_M}, t^{ND_M}, l^{ND_M}, r^{ND_M}, type^{ND_M} \},$$
$$j = \overline{1, J^V_{ND_M}}, k = \overline{1, K^E_{ND_M}},$$

где

ND_M – номер диаграммы модели M;

$J^V_{ND_M}$ – общее количество вершин графа диаграммы $d^M_{ND_M}$;

$K^E_{ND_M}$ – общее количество ребер графа диаграммы $d^M_{ND_M}$;

$\mathrm{v}^{ND_M}_j$ – вершина графа диаграммы $d^M_{ND_M}$;

$\mathrm{e}^{ND_{CE}}_j$ – вершина графа диаграммы $d^M_{ND_M}$;

$s^{ND_M} : E^{ND_M} \to V^{ND_M}$ – функция, ставящая в соответствие каждому ребру $\mathrm{e}^{ND_M}_i$ начальную вершину $\mathrm{v}^{ND_M}_h$;

$t^{ND_M} : E^{ND_M} \to V^{ND_M}$ – функция, ставящая в соответствие каждому ребру $\mathrm{e}^{ND_M}_i$ конечную вершину $\mathrm{v}^{ND_M}_t$;

$l^{ND_M} = (vl^{ND_M}: \mathrm{V}^{ND_M} \rightarrow \mathrm{VL}^{ND_M}, el^{ND_M}: \mathrm{E}^{ND_M} \rightarrow \mathrm{EL}^{ND_M})$ – функции расстановки меток вершинам и ребрам соответственно;

$l^{ND_M} = (vl^{ND_M}: \mathrm{V}^{ND_M} \rightarrow \mathrm{VL}^{ND_M}, el^{ND_M}: \mathrm{E}^{ND_M} \rightarrow \mathrm{EL}^{ND_M})$ – функции расстановки ролей вершинам и ребрам соответственно;

$l^{ND_M} = (vl^{ND_M}: \mathrm{V}^{ND_M} \rightarrow \mathrm{VL}^{ND_M}, el^{ND_M}: \mathrm{E}^{ND_M} \rightarrow \mathrm{EL}^{ND_M})$ – функции расстановки типов вершинам и ребрам соответственно.

Для представления диаграмм подклассы метакласса Relationship метамодели UML представим в виде типов ребер, прямых или косвенных потомков метакласса ModelElement – в виде типов вершин.

Диаграмма классов модели уровня данных $d^{CE}_{ND_{CE}} = D_{ND_{CE}}$.

Определим типы для $D_{ND_{CE}}$:

$\mathrm{VT}^{ND_{CE}}_{\mathrm{name}}$ = {$\mathrm{class}_{\mathrm{entity}}$, attribute, operation};

$\mathrm{ET}^{ND_{CE}}_{\mathrm{name}}$ = {association, dependency, instantiation, ownership}.

Association – тип ребра для описания связи двух классов уровня данных. *Association:* $class_{\mathrm{entity}} \rightarrow class_{\mathrm{entity}}$.

Dependency – тип ребра для описания связи вида «зависимость» от класса уровня данных к классу уровня доступа к данным. *Dependency:* $class_{\mathrm{entity}} \rightarrow class_{\mathrm{data}}$.

Instantiation – тип ребра для описания типов атрибутов и возвращаемых значений операций. *Instantiation: attribute* → $\mathrm{class}_{\mathrm{entity}}$, *Instantiation: operation* → $\mathrm{class}_{\mathrm{entity}}$.

Ownership – тип ребра для описания атрибутов и операций класса. *Ownership: attribute* → $class_{\mathrm{entity}}$, *Ownership: operation* → $class_{\mathrm{entity}}$.

Диаграмма классов модели уровня доступа к данным $d^{CD}_{ND_{CD}} = D_{ND_{CD}}$.

Определим типы для $D_{ND_{CD}}$:

$\mathrm{VT}^{ND_{CD}}_{\mathrm{name}}$ = {$\mathrm{class}_{\mathrm{data}}$, attribute, operation};

$\mathrm{ET}^{ND_{CD}}_{\mathrm{name}}$ ={association, generalization, ownership, instantiation}.

Association – тип ребра для описания связи двух классов уровня доступа к данным. *Association:* $class_{\mathrm{data}} \rightarrow class_{\mathrm{data}}$.

Generalization – тип ребра для описания связи вида «общее-частное» двух классов уровня доступа к данным. *Generalization:* $class_{\mathrm{data}} \rightarrow class_{\mathrm{data}}$.

Instantiation – тип ребра для описания типов атрибутов и возвращаемых значений операций. *Instantiation: attribute* → $class_{\mathrm{data}}$, *Instantiation: operation* → $class_{\mathrm{data}}$.

Ownership – тип ребра для описания атрибутов и операций класса. *Ownership: attribute* → $class_{\text{data}}$, *Ownership: operation* → $class_{\text{data}}$.

Диаграмма классов модели уровня модели пользовательского интерфейса $d^{CM}_{ND_{CM}} = D_{ND_{CM}}$.

Определим типы для $D_{ND_{CM}}$:

$VT^{ND_{CM}}_{name}$ = {$\text{class}_{\text{model}}$, $\text{interface}_{\text{model}}$, attribute, operation};

$ET^{ND_{CM}}_{name}$ = {generalization, dependency, realization, ownership, instantiation}.

Generalization – тип ребра для описания связи вида «общее-частное» двух классов уровня пользовательского интерфейса. *Generalization:* $class_{\text{model}} \rightarrow class_{\text{model}}$.

Dependency – тип ребра для описания связи вида «зависимость» между двумя моделями пользовательского интерфейса, от класса уровня модели интерфейса к классу уровня сервисов, от класса уровня модели интерфейса к классу сервиса. *Dependency:* $class_{\text{model}} \rightarrow class_{\text{model}}$, *Dependency:* $class_{\text{model}} \rightarrow class_{\text{service}}$, *Dependency:* $class_{\text{model}} \rightarrow class_{\text{data}}$.

Realization – тип ребра для описания связи между классом и интерфейсом, при котором класс гарантирует выполнение обязательств интерфейса. Realization*:* $class_{\text{model}} \rightarrow interface_{\text{model}}$.

Instantiation – тип ребра для описания типов атрибутов и возвращаемых значений операций. *Instantiation: attribute* → $class_{\text{model}}$, *Instantiation: operation* → $class_{\text{model}}$, *Instantiation: operation* → $interface_{\text{model}}$.

Ownership – тип ребра для описания атрибутов и операций класса. *Ownership: attribute* → $class_{\text{model}}$, *Ownership: operation* → $class_{\text{model}}$, *Ownership: operation* → $interface_{\text{model}}$.

Диаграмма компонентов интерфейса пользователя $d^{CV}_{ND_{CV}} = D_{ND_{CV}}$.

Определим типы для $D_{ND_{CV}}$:

$VT^{ND_{CV}}_{name}$ = {$\text{component}_{\text{view}}$};

$ET^{ND_{CV}}_{name}$ ={include}.

Include – тип ребра для описания включения элементов пользовательского интерфейса в составной компонент.

Include: $component_{\text{view}} \rightarrow component_{\text{view}}$.

Диаграмма классов контроллеров $d^{CC}_{ND_{CC}} = D_{ND_{CC}}$.

Определим типы для $D_{ND_{CC}}$:

$VT^{ND_{CC}}_{name}$ = {$\text{class}_{\text{controller}}$, $\text{interface}_{\text{controller}}$, operation};

$ET^{ND_{CC}}_{name}$ = {generalization, dependency, realization, ownership, instantiation};

Generalization – тип ребра для описания связи вида «общее-частное» двух классов контроллеров. *Generalization:* $class_{\text{controller}} \rightarrow class_{\text{controller}}$.

Dependency – тип ребра для описания связи вида «зависимость» от класса контроллера к классу сервиса. *Dependency:* $class_{\text{controller}} \rightarrow class_{\text{service}}$.

Realization – тип ребра для описания связи между классом и интерфейсом, при котором класс гарантирует выполнение обязательств интерфейса. *Realization:* $class_{\text{controller}} \rightarrow interface_{\text{controller}}$.

Instantiation – тип ребра для описания типов атрибутов и возвращаемых значений операций. *Instantiation: Instantiation: operation* $\rightarrow class_{\text{controller}}$, *Instantiation: operation* $\rightarrow interface_{\text{controller}}$.

Ownership – тип ребра для описания атрибутов и операций класса. *Ownership: Ownership: operation* $\rightarrow class_{\text{controller}}$, *Ownership: operation* $\rightarrow interface_{\text{service}}$.

Диаграмма классов модели сервисов $d^{CS}_{ND_{CS}} = D_{ND_{CS}}$.

Определим типы для $D_{ND_{CS}}$:

$VT^{ND_{CS}}_{name}$ = {$\text{class}_{\text{service}}$, $\text{interface}_{\text{service}}$, attribute, operation};

$ET^{ND_{CS}}_{name}$ = {generalization, dependency, realization, ownership, instantiation}.

Generalization – тип ребра для описания связи вида «общее-частное» двух классов уровня пользовательского интерфейса. *Generalization:* $class_{\text{service}} \rightarrow class_{\text{service}}$.

Dependency – тип ребра для описания связи вида «зависимость» между двумя сервисами, от класса сервиса к классу сущности, от класса сервиса к классу уровня доступа к данным. *Dependency:* $class_{\text{service}} \rightarrow class_{\text{service,}}$ *Dependency:* $class_{\text{service}} \rightarrow class_{\text{entity,}}$ *Dependency:* $class_{\text{service}} \rightarrow class_{\text{data}}$.

Realization – тип ребра для описания связи между классом и интерфейсом, при котором класс гарантирует выполнение обязательств интерфейса. *Realization:* $class_{\text{service}} \rightarrow interface_{\text{service}}$.

Instantiation – тип ребра для описания типов атрибутов и возвращаемых значений операций. *Instantiation: attribute* $\rightarrow class_{\text{service}}$,

Instantiation: operation → $class_{\text{service}}$, *Instantiation: operation* → $interface_{\text{service}}$.

Ownership – тип ребра для описания атрибутов и операций класса. *Ownership: attribute* → $class_{\text{service}}$, *Ownership: operation* → $class_{\text{service}}$, *Ownership: operation* → $interface_{\text{service}}$.

Диаграмма вариантов использования модели навигационных структур $d^{UC}_{ND_{UC}} = D_{ND_{UC}}$.

Определим типы для $D_{ND_{UC}}$:

$VT^{ND_{UC}}_{name}$ = {useCase, actor};

$ET^{ND_{UC}}_{name}$ = {association, generalization, include, extend} .

Association – тип ребра для описания связи актора и варианта использования. *Association: actor* → *useCase*.

Generalization – тип ребра для описания связи вида «общее-частное» двух акторов или двух вариантов использования. *Generalization: actor* → *actor*, *Generalization: useCase* → *useCase*.

Include – тип ребра для описания связи вида «включение» двух вариантов использования. *Include: useCase* → *useCase*.

Extend – тип ребра для описания связи вида «расширение» двух вариантов использования. *Extend: useCase* → *useCase*.

Диаграмма конечных автоматов модели детализации навигационных сценариев $d^{SM}_{ND_{SM}} = D_{ND_{SM}}$.

Определим типы для $D_{ND_{SM}}$:

$VT^{ND_{SM}}_{name}$ = {state};

$ET^{ND_{SM}}_{name}$ ={transition};

Transition – тип ребра для описания перехода от одного состояния к другому. *Transition: state* → *state*.

Роли и метки

Для вершин графа диаграмм компонентов интерфейса пользователя $d^{CV}_{ND_{CV}}$ $\text{component}_{\text{view}}$ введем роли {actionState, passivState}.

Для вершин графа диаграмм компонентов интерфейса пользователя $d^{CV}_{ND_{CV}}$ с ролью activeState введем метки задания соответствия вершин $\text{component}_{\text{view}}$ с соответствующим ребром transition диаграммы конечных автоматов $d^{SM}_{ND_{SM}}$.

Для вершин графа диаграмм вариантов использования $d^{UC}_{ND_{UC}}$ с типом useCase введем метки задания соответствия с соответствующей диаграммой конечных автоматов $d^{SM}_{ND_{SM}}$.

Для вершин графа диаграмм конечных автоматов $d^{SM}_{ND_{SM}}$ введем роли {frontEndState, serviceState}.

Для вершин графа диаграмм конечных автоматов $d^{SM}_{ND_{SM}}$ с ролью serviceState введем метки задания соответствия вершин состояний с классами диаграммы классов контроллеров $d^{CC}_{ND_{CC}}$.

Для вершин графа диаграмм конечных автоматов $d^{SM}_{ND_{SM}}$ с ролью frontEndState введем метки задания соответствия вершин состояний с классами диаграммы классов интерфейса пользователя $d^{CV}_{ND_{CV}}$.

Процессный граф событийно-ориентированного приложения

Описание событийно-ориентированной модели действий в RIA(Rich Internet Applications)-приложениях основывается на объединениях поведенческих операторов. События определяют системные процессы на определенном уровне абстракции и могут делиться на видимые и невидимые. Множеству видимых событий присвоим метки из множества Act_V, символом τ обозначим любое невидимое событие. Одного символа достаточно τ, поскольку с точки зрения внешнего наблюдателя нет никакого различия невидимых событий. Тогда через $Act = Act_V \cup \tau$ выразим множество всех меток событий приложения. Поведенческие операторы отражают фундаментальные понятия, такие как последовательное соединение, альтернативная композиция и параллельная композиция процессов, представляющих поведение систем. Рассмотрим более подробно виды поведенческих операторов.

Неактивное событие Ø описывает завершившийся процесс. Содержит только одну вершину, нет ребер.

Оператор префикса события *a.P* описывает процесс, который может выполнять *a* и затем ведет себя, как *P*. Этот оператор отображает последовательную композицию событий. Представляется в виде вершины V_0, являющейся корнем для *a.P* и ребра e($V^{a.P}$, V^{P}).

Оператор альтернативной композиции *P1 + P2* представляет собой процесс, который ведет себя и как *P1*, и как *P2* в зависимости от того, какое событие наступит раньше. Если несколько событий могут быть выполнены одновременно, выбор среди них решается недетерминированно из-за отсутствия связанной с ними информа-

ции. Выбор может быть полностью внутренний, если выполняемые события все невидимые, или выбор будет зависеть от воздействия внешней среды. Представляется в виде вершины V_0^{P1+P2}, являющейся корнем для *P1* + *P2* и множества ребер $e(V^{P1+P2}, V^N)$ для каждого ребра $e(V^{P1}, V^N)$ и $e(V^{P2}, V^N)$

Оператор параллельной композиции *P1* ‖ *P2* определяет процесс, который ведет себя поочередно то как *P1*, то как *P2*, с применением множества синхронизованных событий с метками: $S \subseteq Act_V$. Событие, не входящее в множество *S*, выполняется автономно. Синхронизация выполняется принудительно между любыми событиями *P1* и *P2*, имеющимися в множестве *S*, и в этом случае результирующее событие имеет такое же имя, как у двух исходных. Когда *S* = 0, *P1* и *P2* могут выполняться независимо друг от друга. Когда $S = Act_V$, *P1* и *P2* должны синхронизироваться по всем видимым событиям. Представляется в виде пары начальных вершин (V_0^{P1}, V_0^{P2}), являющейся корнем для *P1* ‖ *P2* , множества вершин вида (V^{P1}, V^{P2}), множества ребер $e((V^{P1}, V^{P2}),(V^N, V^{P2}))$ если для *P1* определено ребро $e(V^{P1}, V^N)$, $e((V^{P1}, V^{P2}),(V^{P1}, V^N))$ если для *P2* определено ребро $e(V^{P2}, V^N)$, $e((V^{P1}, V^{P2}),(V^M, V^N))$ если для *P1* определено ребро $e(V^{P1}, V^M)$ и для *P2* определено ребро $e(V^{P2}, V^N)$.

Оператор сокрытия *P* / *H* представляет собой процесс, который ведет себя как *P*, в котором каждое событие с меткой $H \subseteq Act_V$ трансформируется в τ. Этот оператор определяет механизм абстракции по отношению к определенным действиям и может быть использован для предотвращения взаимодействия процесса с внешней средой. Представляется в виде графа путем изменения всех меток ребер графа *P*, принадлежащих множеству *H* на τ.

Оператор ограничения *P* \ *L* представляет собой процесс, который ведет себя как *P*, в котором каждое событие с меткой $L \subseteq Act_V$ предотвращается от выполнения. Результат этого оператора такой же, как результат оператора параллельной композиции с Ø, в которых *L* используется в качестве множества синхронизованных событий. Представляется в виде графа путем удаления из графа *P* всех ребер с метками из множества *L*.

Оператор переименования: *P*[φ] представляет собой процесс, который ведет себя как *P*, в котором каждое событие переименовано в соответствии с общей функцией переименования φ: *Act* →*Act*, сохраняющей видимость действий, т. е. $\varphi^{-1}(\tau) = \{\tau\}$.

Обозначим через *Relab* множество таких функций переименования. Для удобства одиночное переименование можно записать $a \to b$, что означает событие, а переименовывается в b. Представляется в виде графа путем изменения всех меток ребер графа P в соответствии с функцией φ.

Рекурсия: *rec X* : *P* представляет собой процесс, который ведет себя как *P*, в котором каждое свободное вхождение переменной процесса *X* заменяется на *rec X* : *P*. Рекурсивная функция называется конечной, если для всех ее переменных процесс вхождения ограничен, в противном случае она считается бесконечной.

Выводы

Применение модельно-ориентированного подхода для разработки веб-приложений, использование декларативного языка спецификации пользовательского интерфейса и средств автоматической генерации приведет к снижению стоимости и времени разработки. Модельно-ориентированный подход к разработке веб-приложения, предоставляющий автоматическую генерацию интерфейса по декларативным, высокоуровневым моделям позволит ослабить технологическую привязку разрабатываемого интерфейса к конкретной платформе. Однако сам процесс разработки правил трансформации требует формализации моделей для применения автоматической верификации и валидации механизмов трансформации. В работе предложена формализованная метамодель веб-приложения, рассмотрены некоторые аспекты разработки процессного графа событийно-ориентированного приложения, к которым относятся веб-приложения класса Rich Internet Application (RIA). Аппарат теории графов предоставляет средства для формализованного описания метамодели взаимодействия и обеспечивает предоставление точных математических зависимостей между отдельными ее компонентами.

Примечания

[1] См.: *Пупыкина А.А.* Анализ современных подходов к автоматизации разработки веб-приложений // Межотраслевая информационная служба № 3. М.: ВИМИ, 2011. С. 12–20.

[2] *Cunningham W., Beck K.* A Diagram for Object-Oriented Programs [Электронный ресурс] // The ACM Digital Library. URL: http://dl.acm.org/citation.cfm?id=28734 (дата обращения: 07.06.2008).

А.С. Платонова

ПОСТРОЕНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ МНОГОПАРАМЕТРИЧЕСКОГО КОНТРОЛЯ ОБРАЗОВАТЕЛЬНОЙ ДЕЯТЕЛЬНОСТИ

Одним из этапов построения любой информационной системы является разработка архитектуры, пользовательского интерфейса и программ для возможности практического использования всех модулей системы. Выбор технических решений при построении информационной системы многопараметрического контроля образовательной деятельности произведен с учетом специфики конкретной предметной области – контрольно-оценочной деятельности в образовании и необходимости проведения многопараметрического мониторинга на уровне субъекта РФ. В результате созданы структурная схема и пользовательский интерфейс и программно реализованы алгоритмы контроля и формирования многопараметрической оценки учащегося.

Ключевые слова: информационная система, контроль, образовательная деятельность, архитектура, пользовательский интерфейс, программная реализация.

Структурная схема информационной системы многопараметрического контроля образовательной деятельности

Для реализации информационной системы контроля результатов освоения основной образовательной программы в рамках повышения качества школьного образования были разработаны модели описания информационных процессов в системе, математическая модель системы, база данных для хранения результатов контроля, и процедурная модель формирования многопараметрической оценки учащегося. Контроль образовательной деятельно-

сти ведется по шестидесяти параметрам, относящимся к одной из восьми групп.

1. Усвоение теоретического материала.
2. Решение задач.
3. Выполнение лабораторных работ.
4. Выполнение творческих работ.
5. Формирование общеучебных умений и навыков.
6. Развитие.
7. Воспитание.

Мониторинг по 1, 2, 6 группам параметров представляет собой выполнение в системе тестовых заданий самими учащимися. По параметрам из 3, 4, 5 и 7 групп контроль осуществляют учителя: они наблюдают за учащимися в ходе выполнения учебных заданий, делая в системе соответствующие пометки. Результатом работы информационной системы (ИС) является возможность в любое время сгенерировать текущую многопараметрическую оценку учащегося, построить сводные таблицы и графики.

С целью грамотного создания информационной системы, эффективно и надежно функционирующей, построена ее структурная схема и выбрана технология передачи данных. Концепция, определяющая модель, структуру, выполняемые функции и взаимосвязь компонентов информационной системы между собой и с окружением называется архитектурой информационной системы[1]. Конструктивно архитектура обычно определяется как набор ответов на следующие вопросы.

1. Что делает система?
2. На какие части она разделяется?
3. Как эти части взаимодействуют?
4. Где эти части размещены?

При построении программных средств, ориентированных на работу в коммуникационных сетях, необходимо определиться с технологией передачи данных. На сегодняшний день выделяют три основные технологии передачи данных.

1. *Файл-сервер*. В этом случае сервер, на котором находится база данных, является исключительно хранилищем и не обладает каким-либо функционалом, позволяющим производить математические и/или логические вычисления. Данная технология применима исключительно при работе с небольшими объемами данных.

2. *Клиент-сервер*. При использовании клиент-серверной технологии на самом сервере, содержащем базу данных, функционирует некоторое программное обеспечение. Оно называется «Сервером баз данных» и может осуществлять некоторые операции над ин-

формацией, хранящейся в базе данных. Технология клиент-сервер адаптирована для работы с большими объемами данных.

3. *Терминал-сервер*. Принципиально отличается от двух предыдущих технологий тем, что конечному пользователю по сети передаются не сами интересующие его данных, а изображение этих данных. Фактически пользователь работает за другим компьютером, физически удаленным от него, получая по сети только изображение Рабочего стола с запущенными программами. Имеет низкую скорость обмена информацией при работе через глобальную сеть[2].

Для обеспечения доступа всех участников контрольно-оценочной деятельности к единой базе данных будем строить информационную систему в рамках технологии клиент-сервер. Клиент-сервер – это технология (архитектура) взаимодействия клиента и сервера. Клиент – программа, запрашивающая у сервера информацию или выполнение какого-либо задания на сервере от имени клиента. Сервер – это прикладная программа, исполняющая запросы клиента. Клиент и сервер взаимодействуют по определенному протоколу. Программа клиента и программа сервера могут располагаться как на одной машине, так и на совершенно различных компьютерах произвольной сети[3].

С точки зрения количества составных частей клиент-серверные системы делятся на двухуровневые и трехуровневые. Двухуровневые системы состоят только из клиента и сервера. В трехуровневых же между пользовательским клиентом и сервером, осуществляющим хранение и обработку базы данных, появляется третий промежуточный слой, являющийся для пользователя сервером, а для системы управления базами данных – клиентом[4]. Это позволяет более гибко распределять функции системы и нагрузку между компонентами программно-аппаратного комплекса, а также может снизить требования к ресурсам рабочих мест пользователей. Необходимой платой за это является то, что подобные системы намного сложнее в разработке, внедрении и эксплуатации и требуют значительных затрат и высококвалифицированного персонала[5].

Таким образом, структурная схема разрабатываемой информационной системы построена на основе технологии клиент-сервер и имеет трехуровневую архитектуру (рис. 1). Это позволяет изолировать уровни друг от друга, быстро и простыми средствами переконфигурировать систему при возникновении сбоев или при плановом обслуживании на одном из уровней. Компоненты архитектуры клиент-сервер, с точки зрения программного обеспечения, реализуют сервер базы данных, сервер приложений и браузеры. Взаимодействие между сервером базы данных и браузером осуществляется

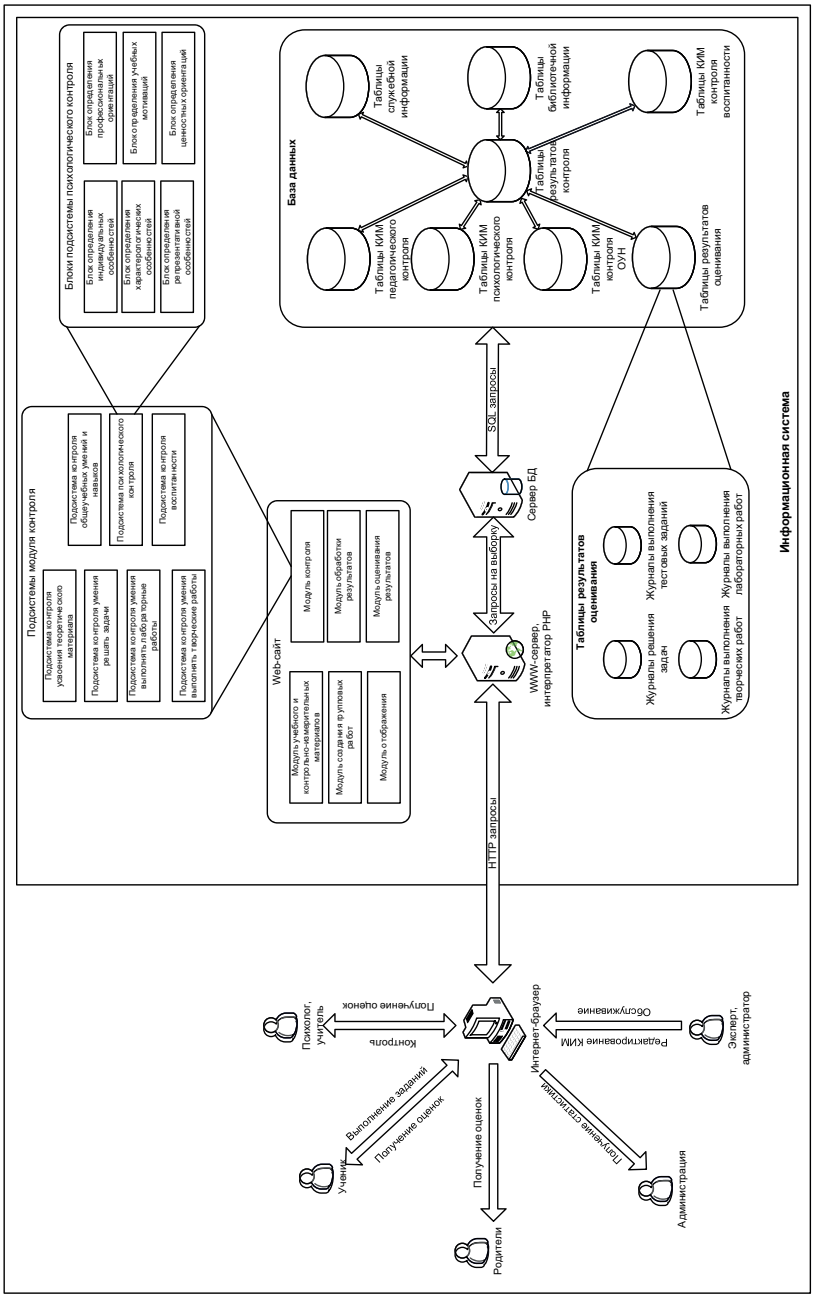


Рис. 1. Структурная схема информационной системы

через глобальную сеть посредством web-сервера. Сервер базы данных, осуществляющий хранение и обработку базы данных, представлен MySQL-сервером; сервер приложений – web-сервером Apache и web-сайтом, написанном с помощью PHP-скриптов; роль клиента выполняет любой web-браузер.

В основе разрабатываемой информационной системы лежит база данных. База данных содержит учебные материалы, контрольно-измерительные материалы, библиотечную, справочную и служебную информацию, итоговые и промежуточные результаты контроля в виде баллов. Пользователи посредством программы-браузера, через web-сервер, обращаются к содержанию базы данных. Форма содержания зависит от типа пользователя и выполняемой им задачи (прохождение теста, дополнение контрольно-измерительных материалов, получение информации об учащемся и т. п.). В структуру разработанного сайта входят несколько модулей, в том числе модуль контроля, обеспечивающий контроль по группам параметров с помощью набора контролирующих подсистем.

Пользовательский web-интерфейс и программная реализация информационной системы многопараметрического контроля образовательной деятельности

Для создания информационной системы с использованием web-технологий разработано достаточное количество языков web-программирования, основными из которых являются следующие:

1. *JavaScript*. Скрипты, написанные на JavaScript, выполняются на стороне клиента (т. е. на компьютере пользователя), и не просто на стороне клиента, а в самом браузере, поэтому не требуется никаких дополнительных программ, никаких плагинов и т. п., скрипт можно вставить в html-код страницы, и он будет выполняться в любом браузере.

2. *Java*. В отличие от JavaScript, программа на Java не встраивается в html-код, а работает под управлением специальной виртуальной машины Java. Кроме этого, на языке Java можно создавать Java-апплеты, маленькие автономные программы, которые можно вызвать в html-коде и выполнение которых обеспечивается браузером.

3. *PHP*. Благодаря своей простоте и гибкости быстро собрал множество поклонников по всему земному шару и стал одним из наиболее популярных языков веб-программирования. Как и в слу-

чае с JavaScript, код PHP можно писать вперемешку с html-кодом, с той лишь разницей, что этот код выполняется сервером до отправки страницы клиенту и в браузер попадает лишь результат работы скрипта. При своей простоте и удобстве использования PHP весьма универсален, с легкостью позволяет писать программы, работающие с самыми различными базами данных, с графикой и т. д.[6]

4. *Ajax*. В последнее время набирает популярность. Самым большим плюсом использования технологии Ajax является возможность связываться с web-сервером без перезагрузки самой web-страницы. Это создает впечатление беспрерывности работы сайта. Такая возможность пересылки и приема данных из сервера без обновления всей страницы возможна благодаря использованию объекта XMLHttpRequest, который является центральной частью всей технологии Ajax[7].

5. *Perl*. Один из популярных языков для сетевого программирования, в частности создания веб-сайтов. И на сегодняшний день Perl стал неотъемлемым инструментом в инструментарии web-программистов. В отличие от таких технологий как JavaScript и VBScript, perl-скрипты выполняются на сервере, что позволяет создавать полноценные интерактивные приложения, управлять базами данных, почтой, создавать баннерные сети, счетчики, гостевые книги, форумы и многое другое[8].

Для реализации взаимодействия пользователя с контентом сайта разработанной ИС был использован PHP. А для взаимодействия с сервером базы данных MySQL применялся язык SQL-запросов. SQL-язык, который дает вам возможность создавать и работать в реляционных базах данных, которые являются наборами связанной информации, сохраняемой в таблицах[9]. В настоящее время язык SQL поддерживается многими десятками СУБД различных типов, разработанных для самых разнообразных вычислительных платформ. Рассматриваемый язык SQL ориентирован на операции с данными, представленными в виде логически взаимосвязанных совокупностей таблиц отношений. Важнейшая особенность его структур – ориентация на конечный результат обработки данных, а не на процедуру этой обработки[10].

В рамках программной реализации процедурных моделей был реализован пользовательский web-интерфейс. Интерфейс – совокупность средств и правил, обеспечивающих взаимодействие устройств вычислительной системы и программ, а также взаимодействие их с человеком. С ростом сети Internet широкое распространение получили web-интерфейсы, позволяющие взаимодействовать с различными программами через браузер, который

включает в себя основное меню, набор ссылок, реализующих те или иные функции системы и центральный блок, отображающий текущую задачу[11].

Пользовательский интерфейс был разработан на основе web-шаблона Lightneasy с помощью PHP и JavaScript[12].

Итак, набор возможностей разрабатываемого интерфейса зависит от пользователя, прошедшего аутентификацию. Для работы в системе пользователь должен авторизоваться (ввести свой логин и пароль) или в случае нового пользователя заполнить форму регистрации (рис. 2).

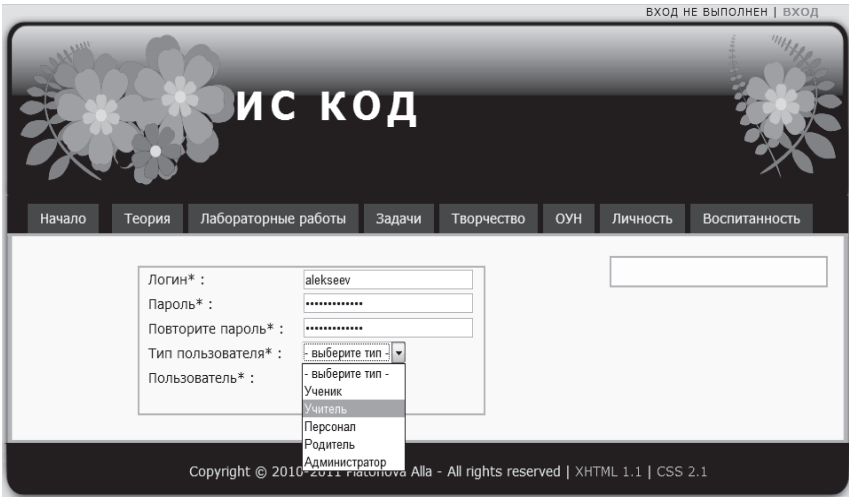


Рис. 2. Регистрация пользователя

Возможности (набор функций) администратора и эксперта, учителя, ученика, родителей, административного персонала или гостя отличаются друг от друга. Непосредственно выполнению пользователями тех или иных задач предшествует экран с инструкцией о данной функции.

Интерфейс администратора и эксперта включает в себя следующие задачи:

– управление пользователями (активация, редактирование и удаление аккаунтов пользователя) (рис. 3);

– функции редактирования учебного контента (добавление и редактирование тестовых заданий, методик и т. п.) (рис. 4)[13].

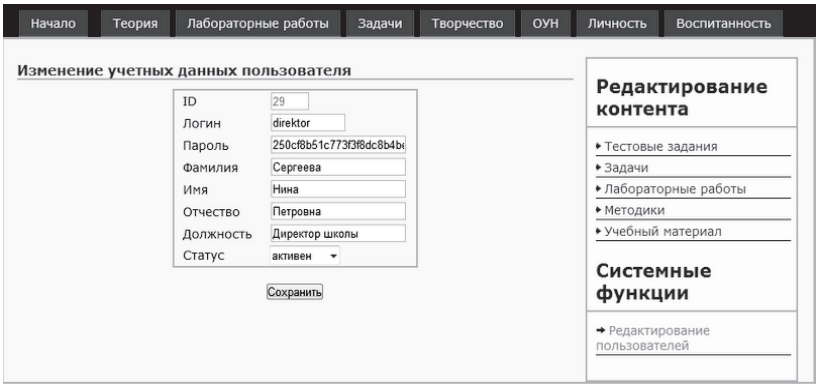


Рис. 3. Редактирование пользователя

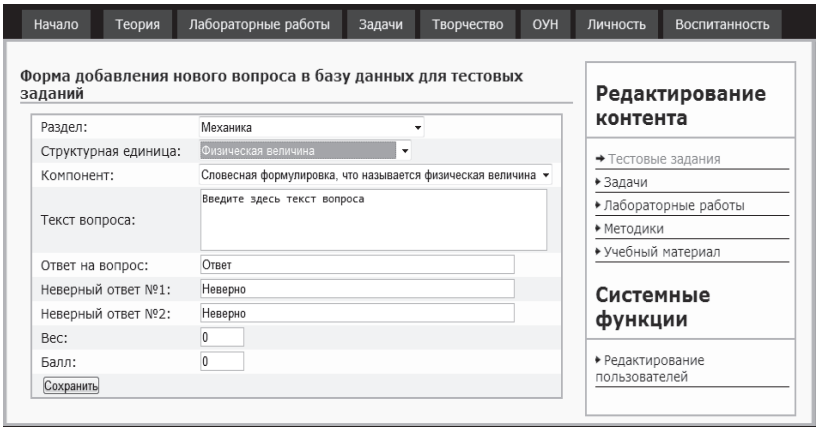


Рис. 4. Редактирование учебного материала

Интерфейс учителя включает в себя следующие задачи:

– контроль умения выполнять лабораторные и творческие работы, сформированность общеучебных умений и навыков, и воспитанности учащихся (фактически это заполнение учителями электронных таблиц контроля) (рис. 5);

– формирование оценок: генерация многопараметрической оценки достигнутого уровня результатов образования любого из учащихся или предоставление статистической информации в виде различного вида графиков или таблиц.

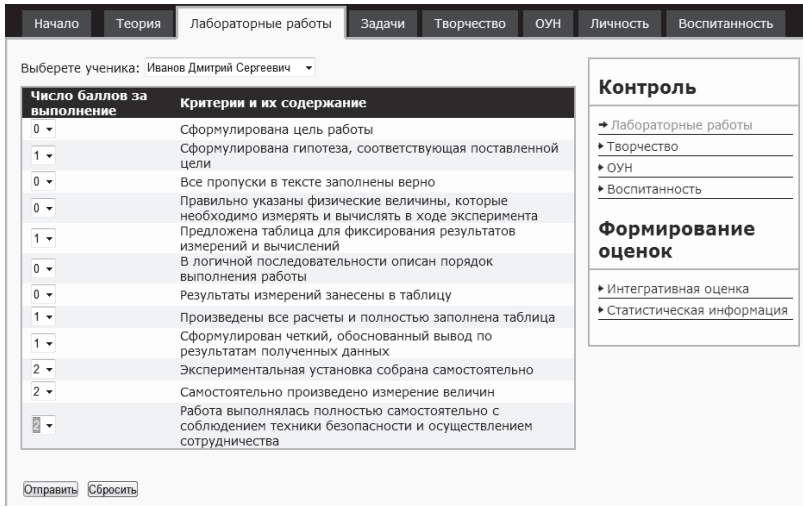


Рис. 5. Форма оценивания лабораторной работы

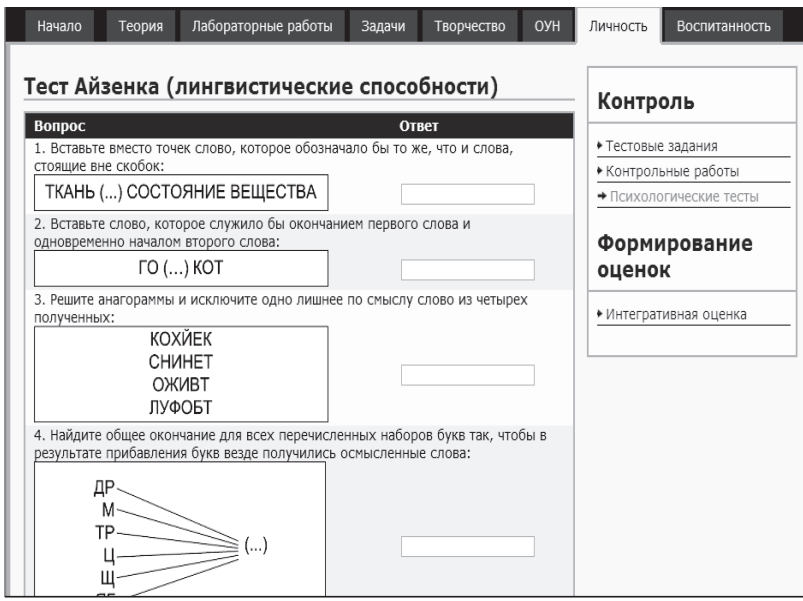


Рис. 6. Психологический контроль

Интерфейс ученика предоставляет следующие возможности:

– выполнение тестовых заданий на проверку усвоения теоретического материала, решение контрольной работы, состоящей из нескольких задач, выполнение психологических тестов (рис. 6);

– формирование оценок, в частности генерирование многопараметрической оценки достигнутого уровня своих результатов образования.

Интерфейс родителей включает в себя задачу получения многопараметрической оценки достигнутого уровня результатов образования своего ребенка.

Интерфейс административного персонала (городские, районные и областные управления образования) позволит решать задачи предоставления многопараметрических оценок учащихся и статистической информации по ученику, классу, параллели, школе, городу и области (рис. 7).

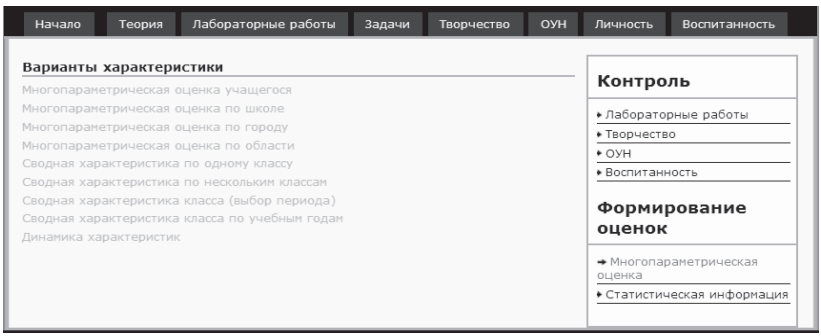


Рис. 7. Интерфейс администрации

Программная реализация разработанных процедурных моделей заключается в использовании в теле HTML-страницы структур языка PHP и SQL-запросов[14]. Несколько повторяющихся алгоритмических последовательностей (например, извлечение списка класса) выполнены в виде функций, которые можно вызывать неоднократно и с разным набором входных переменных. Кроме того, некоторые процедуры, в частности обработка результатов тестирования, выполняется исполняемым PHP-скриптом в чистом виде без использования HTML[15].

Результат программной реализации процедуры выбора и отображения формы с вопросами тестовых заданий выглядит следую-

щим образом (рис. 8). После нажатия кнопки завершения тестирования управление переходит к PHP-скрипту post.php, указанному в описании формы, в котором происходит обработка ответов учащегося. Ответы ученика и дополнительные переменные передаются PHP-скрипту через HTTP-запрос.

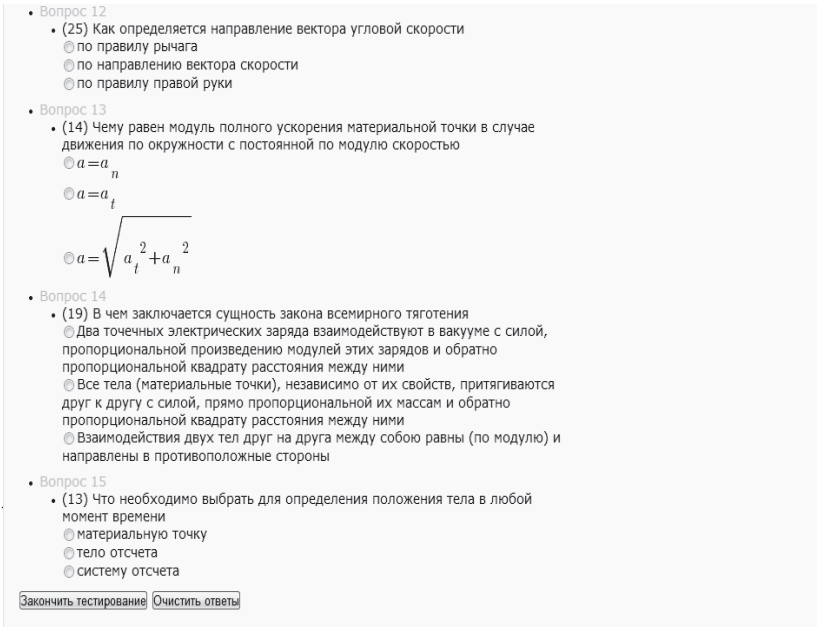


Рис. 8. Выполнение учеником тестового задания для контроля усвоения теоретического материала

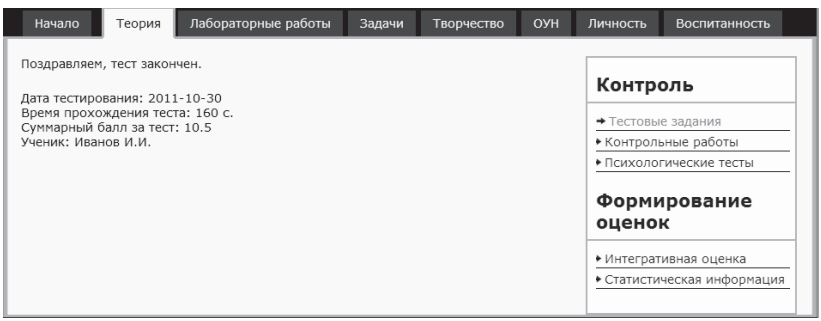


Рис. 9. Результат выполнения тестовых заданий

Таким образом, была реализована вся процедурная модель формирования многопараметрической оценки учащегося.

При нажатии кнопки «Многопараметрическая оценка» информация, хранящаяся в базе данных, в соответствии с процедурной моделью преобразуется в расширенную психолого-педагогическую информацию о достижениях учащегося и выстраивается в соответствии с разработанным ранее шаблоном. Фрагменты экранной формы самой многопараметрической оценки учащегося представлены на рис. 10 и 11.

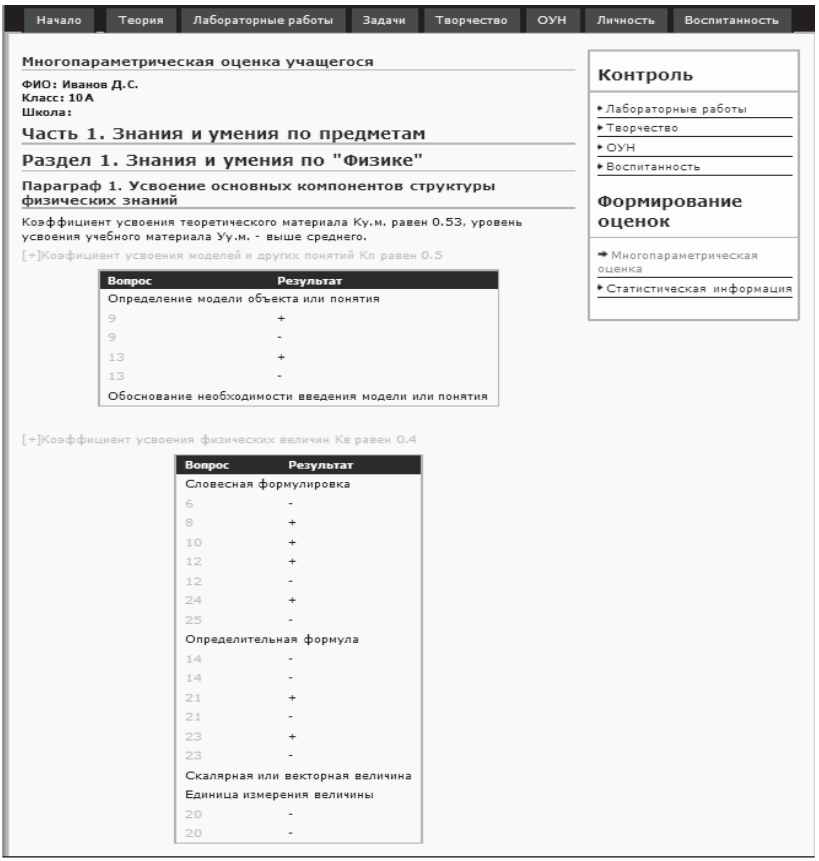


Рис. 10. Фрагмент 1 многопараметрической оценки

Параграф 4. Професиональная ориентация

Тип рекомендуемой профессии П - «Человек – знаковая система».

Подробнее

Профессии, труд в которых направлен на обработку информации (сведений), представленной в виде условных знаков, цифр, формул, текстов:
1. Создание и оформление документов (на родном или иностранном языке), делопроизводство, анализ текстов или их преобразование, перекодирование.

2. Профессии, труд в которых направлен на числа, количественные соотношения.

3. Профессии, труд в которых направлен на системы условных знаков, схематические отображения объектов.

Параграф 5. Учебная мотивация

Мотивы учебной деятельности Му.д. - учебно-позновательные: ориентирующие ученика на усвоение способов добывания знаний, приемов самостоятельного приобретения знаний. Заключаютс в самостоятельных действиях по поиску разных способов решения, в вопросах учителю о сравнении разных способов работы.

Параграф 6. Ценностные ориентации

Уровень по шкале «Познание как ценность» Уц.п. - средний
Учащийся понимает значение образования, воспринимает познание как важ¬ную ценность в жизни, хотя полного осознанного самоопреде¬ления по отношению к этой ценности в сознании не произошло.
Уровень по шкале «Я-ценность» Уц.я. - высокий
У учащегося практически сформировалась структура «Я» под¬ростка, высокая степень позитивного самовосприятия, уме¬ния самовыражаться, уважительного отношении к себе как к лич¬ности и индивидуальности.
Уровень по шкале «Другой-ценность» Уц.д. - низкий
Отсутствие у учащегося восприятия другого человека как ценности и индивидуальности, отсутствие уважительного отношения к окружаю¬щим, недостаток умения общаться с ними доброжелательно и конструктивно.
Уровень по шкале «Общественно полезная деятельность» Уц.об. - средний
Учащийся понимает важность общественно полезной деятель¬ности и готов отдавать ей некоторую часть своего времени и сил, но не всегда это становится результатом его личной иници¬ативы и самостоятельности.
Уровень по шкале «Ответственность как ценность» Уц.от. - низкий
Учащийся в очень малой степени либо вообще не осознает необходимость ответствен-ности человека за его поступки в жизни, за выбор модели поведе¬ния. В этом случае преобладает экстернальный (внешний) контроль в жизненно важных ситуациях. Личность не считает ответственность ценностью, проявляет при этом низкий уровень рефлексии.

Часть 4. Воспитание

Таблица

Проявления	Уровень
Внешний вид, прическа, одежда, украшения и пр., Ув.в.	Высокий
Общая речь, Уо.р.	Высокий
Отношения с ребятами, Уотн.р.	Высокий
Отношения с учителями, Уотн.у.	Высокий
Отношения с другими, Уотн.д.	Высокий
Дисциплинированность в плане выполнения указаний учителя, Уд.	Высокий
Отношение к школьному имуществу, к труду других, Котн.им.	Высокий

Общий балл Бв. равен: 5; уровень воспитанности Ув. - высокий

Рис. 11. Фрагмент 2 многопараметрической оценки

Целью использования в учебном процессе информационной системы многопараметрического контроля образовательной деятельности является повышение уровня предметной подготовки, наиболее полное раскрытие способностей и, наконец, всестороннее развитие личности каждого учащегося. Главным результатом рабо-

ты с системой является получение пользователями многопараметрической оценки учащегося, в которой содержится большой объем психолого-педагогической информации о достижениях учащегося и рекомендации для него. Информационная система является инструментом, дополнительным средством оценивания учащегося наряду с традиционными методами и средствами контроля в школе.

Примечания

[1] См.: Служба тематических толковых словарей [Электронный ресурс]. URL: http://www.glossary.ru (дата обращения: 22.10.2013).

[2] См.: Технологии передачи данных: файл-сервер, клиент-сервер, терминал-сервер [Электронный ресурс] // Компания «Сатурн». URL: http://www.itsaturn.ru/articles/article14.html (дата обращения: 22.10.2013).

[3] См.: *Голицына О.Л., Максимов Н.В.* Информационные системы: Учеб. пособие. М.: 2004. 329 с.

[4] См.: Системы «клиент-сервер». Ч. 2 [Электронный ресурс] // Сайт А.И. Белостоцкого. URL: http://belani.narod.ru/1/Lklser2.htm (дата обращения: 22.10.2013).

[5] См.: Там же.

[6] См.: Веб-программирование (обзорная статья) [Электронный ресурс] // Все о web-дизайне. URL: http://wseweb.ru/diz/obzor3.htm (дата обращения: 22.10.2013).

[7] См.: Преимущества и недостатки технологии Ajax [Электронный ресурс] // Создание и раскрутка сайтов. URL: http://microwebnet.blogspot.com/2011/03/ajax.html (дата обращения: 22.10.2013).

[8] См.: Главная страница Perl [Электронный ресурс]. URL: http://perl.far.ru (дата обращения: 22.10.2013).

[9] См.: *Грабер М.* Понимание SQL [Электронный ресурс]. // SQL. URL: http://www.sql.ru/docs/sql/u_sql (дата обращения: 22.10.2013).

[10] См.: Введение в структурированный язык запросов SQL [Электронный ресурс] // Учи IT! URL: http://www.uchi-it.ru/11/4/1.html (дата обращения: 22.10.2013).

[11] См.: Интерфейс [Электронный ресурс] // your-hosting.ru. URL: http://your-hosting.ru/terms/i/interface (дата обращения: 22.10.2013); Web-интерфейс [Электронный ресурс] // Там же. URL: http://your-hosting.ru/terms/rv/wi (дата обращения: 22.10.2013).

[12] См.: LightNEasy [Электронный ресурс]. URL: http://www.lightneasy.org (дата обращения: 22.10.2013).

[13] См.: *Колдаев В.Д.* Основы алгоритмизации и программирования: Учеб. пособие. М.: ФОРУМ; ИНФРА-М, 2006.

[14] См.: *Кузнецов М., Симдянов И.* MySQL 5. СПб.: БХВ, 2010.

[15] См.: *Прохоренок Н.А.* HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера. СПб.: БХВ, 2011.

Н.Р. Мартынов, О.В. Казарин

ПОДПОРОГОВЫЕ КАНАЛЫ И МЕТОДЫ ЗАЩИТЫ ОТ ИХ СОЗДАНИЯ В СХЕМАХ ИНТЕРАКТИВНОЙ ИДЕНТИФИКАЦИИ

Статья посвящена анализу свойств подпороговых каналов. Показывается возможность существования широкополосных подпороговых каналов в схемах электронной подписи из отечественных стандартов ГОСТ Р 34.10-94, ГОСТ Р 34.10-2001, ГОСТ Р 34.10-2012 и схемах интерактивной идентификации пользователя. Особое внимание уделяется принципам формирования широкополосных подпороговых каналов и методам защиты от их создания.

Подпороговые каналы можно рассматривать и как стеганографические каналы, создаваемые в различных криптографических конструкциях, где их некоторые случайные параметры заменяются на подпороговые сообщения, специально создаваемые потенциальными нарушителями.

Ключевые слова: подпороговый канал, широкополосный подпороговый канал, схема электронной подписи, схема интерактивной идентификации.

Впервые понятие подпорогового канала (subliminal channel) было введено в 1983 г. Г.Дж. Симмонсом[1]. Позднее он показал, что в протоколе электронной подписи Эль-Гамаля возможно введение подпорогового сообщения. Посредством такого канала злоумышленник, например нечестный сотрудник банка, может выдавать своему сообщнику, имеющему доступ к банковским коммуникациям, оперативную финансовую информацию. При этом выдаваемые сообщения имеют на первый взгляд вполне достоверный вид.

Кроме того, при помощи подпорогового канала злоумышленники, получившие доступ к сетевым коммуникациям, могут обмени-

ваться между собой оперативной информацией, которая является недоступной для служб безопасности сети[2].

Подпороговые каналы

Обозначим через S участника протокола электронной подписи, который подписывает сообщения и посылает их вместе с подписями другому участнику, обозначаемому через R. Участник протокола R осуществляет проверку подлинности подписей. Предположим, что S хочет использовать протокол для тайной передачи информации своему сообщнику R' в качестве которого может выступать как R, так и внешний наблюдатель, перехватывающий подписанные сообщения. При этом предполагается, что передаваемая таким образом информация и сам факт такой передачи не могут быть раскрыты с помощью общедоступных сведений. Способ, который позволяет S это сделать с достаточно большой вероятностью, и является *подпороговым каналом*.

В качестве одного из возможных злоупотреблений, основанных на использовании подпороговых каналов, Г.Дж. Симмонс предлагал следующий сценарий. Предположим, что некоторый центр (например, правительственная организация) использует протокол электронной подписи для выдачи документов (удостоверений личности, водительских прав и т. п.). Подпись на таком документе должна содержать только ту информацию, которая свидетельствует, что документ действительно был выдан данным центром. Но при наличии подпорогового канала центр может сообщать своим секретным агентам дополнительную информацию о владельце документа (информацию о благосостоянии, политической неблагонадежности, другие персональные данные).

Возможен и другой сценарий использования подпорогового канала. В некотором коммерческом центре (например, банке) находится компьютер с конфиденциальной информацией. На компьютере работает ограниченный круг допущенных лиц, которые регулярно отправляют подписанные сообщения, скажем, в филиал. Тогда один из допущенных сотрудников может использовать подпороговый канал для передачи конфиденциальной информации своему сообщнику. Здесь важно подчеркнуть, что подпороговый канал позволяет наладить оперативную и регулярную передачу конфиденциальной информации.

Выделяют две разновидности подпороговых каналов. *Широкополосный канал* позволяет передавать вместе с каждым подписан-

ным сообщением длинное (состоящее из большого числа битов) подпороговое сообщение (т. е. сообщение для R'), но, вообще говоря, требует, чтобы R' знал секретный ключ участника S. *Узкополосный канал* не требует знания участником R' секретного ключа участника S, но позволяет передавать лишь короткие подпороговые сообщения (порядка нескольких битов), поскольку для реализации схемы необходимо инвертировать некоторую одностороннюю функцию.

В 1993 г. Г.Дж. Симмонс показал, что в американском алгоритме стандарта на электронную цифровую подпись DSA существует широкополосный подпороговый канал. Дальнейшие рассуждения с незначительными изменениями применимы и для алгоритма отечественного стандарта на электронную подпись ГОСТ Р 34.10-94 (их можно распространить с небольшими изменениями и на стандарты ГОСТ Р 34.10-2001, ГОСТ Р 34.10-2012).

Пусть подпороговые сообщения отождествлены с элементами $Z_q^* \setminus \{1\}$. Чтобы передать подпороговое сообщение $\chi \in Z_q^* \setminus \{1\}$ вместе с подписью для сообщения m (все обозначения, кроме касающихся непосредственно подпорогового канала, даны в соответствии с ГОСТ Р 34.10-94), участник S выбирает k в процедуре выработки подписи не случайным, а равным χ. Если получившиеся при этом $r = f(\chi)$ и $s \equiv [\chi h(m) + xr] (\mathrm{mod}\ q)$ (ГОСТ) или $s \equiv [\chi^{-1}(h(m) + xr)] (\mathrm{mod}\ q)$ (DSA) не равны 0 (что выполняется с высокой вероятностью), то S отправляет сообщение m с подписью (r, s) участнику R. Если R' знает секретный ключ x участника S, то получив (m, r, s), он может найти χ, получив его из формулы для s, т. е. $\chi \equiv (s - xr)h(m)^{-1} (\mathrm{mod}\ q)$ (ГОСТ) или $\chi \equiv [(h(m) + xr)s^{-1}] (\mathrm{mod}\ q)$ (DSA).

Создание широкополосного подпорогового канала в схемах интерактивной идентификации

Создание широкополосного подпорогового канала возможно не только в схемах электронной подписи, но и в схемах интерактивной идентификации пользователя. На примере схем интерактивной идентификации[3] далее показывается, как реализовать подпороговый канал в них.

Схемы интерактивной идентификации, рассматриваемые в данной работе, позволяют обеспечить надежный доступ абонентов некоторой информационной системы к различного рода удаленным ресурсам. В таких системах субъект доступа гарантированно доказывает администратору системы (объекту доступа) наличие у него

секретных реквизитов, не раскрывая никакой сколько-нибудь значимой информации об этих реквизитах.

Данная схема предусматривает наличие третьего необходимого элемента – центра распределения ключей (ЦРК), который осуществляет основные управляющие функции по установлению защищенных соединений в системе.

Описание схемы интерактивной идентификации. Центр распределения ключей выбирает простые числа p, q и r такие, что $qr \mid p - 1$, где размерности этих чисел в битах составляет $|q| \geq 256$, $|r| \geq 256$ и $|p| \geq 1024$, затем находит такое $g \in Z_p$, что $g^q \equiv 1 (\mathrm{mod}\ q)$, $g \neq 1$.

Абонент системы (обозначим его через P) приходит для регистрации в ЦРК и вместе с ЦРК выбирает секретный ключ $x \in Z_q$, а также соответствующий ему открытый ключ $y \equiv g^x (\mathrm{mod}\ p)$. Далее для этого абонента ЦРК составляет идентификационную строку I, состоящую из имени, адреса, уровня полномочий и т. п. Затем для y и I генерируется цифровая подпись ЦРК s. Абоненту выдаются: подпись s, ключи x и y, строка I и простое q.

Процесс интерактивной идентификации, в котором абонент P сначала будет доказывать администратору, что s есть подпись ЦРК, а затем, что он имеет в наличии секретный ключ x без раскрытия самого секретного значения x, начинается с отсылки подписи s и значений I, y администратору V. Администратор V с помощью открытого ключа ЦРК верифицирует идентификационную строку I и открытый ключ y абонента P. Схема подписи в данном случае может быть любой из известных.

Протокол IIP

Интерактивная часть протокола выполняется в l циклах по i:

1. Абонент P выбирает случайное $k_i \in {}_R Z_q$ и вычисляет $r_i \equiv g^{k_i} (\mathrm{mod}\ p)$, значение r_i отсылается администратору.

2. Администратор V выбирает $e_i \in {}_R \{1,...,2^t - 1\}$, где t – некоторый параметр безопасности и выдает e_i абоненту P.

3. Абонент P вычисляет $s_i \equiv [xr_i + k_i e_i] (\mathrm{mod}\ q)$ и отсылает s_i администратору.

4. Администратор V осуществляет контроль: $g^{s_i} \equiv y^{r_i} r_i^{e_i} (\mathrm{mod}\ p)$.

Если проверки на шаге 4 во всех l циклах завершены корректно, то процесс идентификации абонента системы завершен успешно, в противном случае администратор сигнализирует в службу безопасности о попытке несанкционированного доступа.

Примечание 1. Обозначение $k \in_R K$ означает случайный и равновероятный выбор элемента k из всех элементов множества K.

Описание схемы интерактивной идентификации с введенным подпороговым каналом. Пусть P является зарегистрированным абонентом системы. В то же время он может быть и злоумышленником, если пытается использовать параметры схемы интерактивной идентификации для отсылки подпороговых сообщений своему сообщнику.

Соглашения в данном случае точно такие же, как и в предыдущей схеме. Без потери общности, пусть $l = 1$. Предположим, что подпороговые сообщения отождествлены с элементами $Z_q^* \setminus \{1\}$. Чтобы передать подпороговое сообщение z, абонент P выбирает k в процедуре выработки r не случайным, а равным z.

1. Абонент P вычисляет $r \equiv g^z (\mathrm{mod}\ p)$, где z – подпороговое сообщение, и значение r отсылается администратору.

2. Администратор V выбирает $e \in_R \{1, ..., 2^t - 1\}$ выдает его абоненту P.

3. Абонент P вычисляет $s \equiv [xr + ze] (\mathrm{mod}\ q)$ и отсылает s администратору.

Сообщник абонента P, зная секретный ключ x и получив s, подключившись к коммуникациям сети между администратором V и абонентом P, может найти z следующим образом: $z = (s - xr)e^{-1} (\mathrm{mod}\ q)$. В то время как администратор V, выполнив проверку $g^s \equiv y^r r^e (\mathrm{mod}\ p)$, этого не заметит.

Методы защиты от введения подпороговых каналов

Метод основывается на введении в схемы специального программно-технического устройства (специальной программы), именуемого охранным, которое предположительно является доверенным. Такое устройство позволяет защищаться от злоумышленников, которые пытаются использовать случайные параметры схемы для генерации подпороговых сообщений.

Отметим здесь также, что данный метод работает не только против нечестного абонента P и его сообщника, но и против нечестного администратора, который пытается использовать случайную строку запроса для отсылки подпорогового сообщения своему сообщнику.

Обозначим через G абонента системы, функционирующего как охранное устройство, через P и V – честных абонентов системы, а

через P^* и V^* – абонентов, пытающихся послать подпороговые сообщения, своим сообщникам. Все коммуникации от абонента к администратору и обратно проходят через G. Обозначение $\mu^{(\nu)}$ означает, что абонент G сформировал параметр μ вместо параметра n, которым обмениваются абонент и администратор (или, по крайней мере, они подразумевают, что обмениваются ν).

Пусть **Пр** обозначает некоторый универсальный алгоритм (алгоритм противника) для нечестных абонентов системы, посредством которого они пытаются создавать подпороговые сообщения в схеме интерактивной идентификации.

Протокол ИР^G

1. Абонент P вычисляет $r \equiv g^k (\mathrm{mod}\ p)$, где $k \in {}_R Z_q$ и отправляет r абоненту G.

1*. Абонент P^* вычисляет $r \equiv g^k (\mathrm{mod}\ p)$, где $k = \mathbf{Пр}(z')$ и отправляет r абоненту G (пытается обмануть G).

2. Абонент G вычисляет $\alpha \equiv r^d (\mathrm{mod}\ p) \equiv g^{kd} (\mathrm{mod}\ p)$, где $d \in {}_R Z_q$ и отправляет $\alpha^{(r)}$ абоненту P и администратору V.

3. Администратор V по получении $\alpha^{(r)}$ высылает $e \in {}_R\{1, ..., 2^t - 1\}$, где t – некоторый параметр безопасности, абоненту P. По пути e принудительно поступает к абоненту G.

3*. Администратор V^* по получении $\alpha^{(r)}$, используя алгоритм **Пр**, встраивает подпороговое сообщение z'' в e: $e = \mathbf{Пр}(z'')$. По пути e принудительно поступает к абоненту G.

4. Абонент G преобразует: $\beta \equiv ed (\mathrm{mod}\ q)$, после чего b выдается P.

5. Абонент P вычисляет $s \equiv [(x\alpha + k\beta)] (\mathrm{mod}\ q)$ и доказывает абоненту G, что он использовал вместо r и e значения $\alpha^{(r)}$ и $\beta^{(e)}$.

Делает он это (проводит доказательства) посредством выполнения интерактивного протокола доказательства (с нулевым разглашением) равенства $g^s \equiv y^\alpha r^\beta (\mathrm{mod}\ p)$, в основе которого лежит протокол доказательства равенства двух дискретных логарифмов[4].

Если доказательства прошли успешно, тогда абонент P (или абонент G) отсылает s администратору V.

5*. Абонент P^* пытается провести доказательства так, чтобы обмануть G, а затем отправить в значении s какое-либо подпороговое сообщение z'''.

6. Администратор V может проверить: $g^s \equiv y^{\alpha^{(r)}} [\alpha^{(r)}]^e (\mathrm{mod}\ p)$.

Примечание 2. Частью алгоритма **Пр** может быть, например, абсолютно стойкий шифр Вернама. Такой сценарий выглядит вполне

убедительным, так как подпороговые сообщения достаточно короткие (в вышеприведенных схемах они ограничены длиной модуля) и такой ресурсозатратный шифр можно использовать для получения криптограммы для подпорогового сообщения.

Замечания по безопасности схемы. При обсуждении безопасности данной схемы мы оставляем открытыми вопросы доказательства следующих трех утверждений (кроме первого, доказательство которого очевидно).

1) Если абонент P и администратор V являются честными, тогда последний примет доказательства первого с вероятностью 1 (*свойство полноты протокола интерактивной идентификации*).

2) Если абонент P не знает секретного ключа x (т. е. он P^*), то тогда, что бы ни предпринимал P^*, он не сможет обмануть администратора V с вероятностью, близкой к 1 (*свойство корректности протокола*).

3) Если абонент P и администратор V отклоняются от протокола (т. е. они P^* и V^*), тогда по имеющимся a и b :

во-первых, нечестный администратор V (т. е. V^*) не может получить никакой полезной для себя информации о значении x (*свойство нулевого разглашения*[5] *протокола*);

во-вторых, сообщник, прослушивающий каналы связи между абонентами P, G и V, не может получить сколько-нибудь полезной для себя информации о подпороговых сообщениях, генерируемых P^* и V^* (*свойство надежности охранного устройства*).

В целом одним из необходимых условий для доказательства стойкости протокола IIP^G является стойкость протокола IIP как интерактивной системы доказательств с нулевым разглашением, которая, в свою очередь, может быть доказана стандартным образом[6]. Необходимым условием для доказательства свойства защищенности от создания подпорогового канала в этом протоколе также является (физическая) надежность используемого охранного устройства G.

Для доказательства общей стойкости трехстороннего (транзитивного) интерактивного протокола IIP^G для различных моделей противника скорее всего придется использовать более сложный математический аппарат. В данном случае предлагается использовать гибридную модель Канетти – hybrid UC-модель (universally composable model)[7]. Стойкость протоколов в такой гибридной UC-модели доказывается в соответствии с теоремой о стойкости композиции безопасных многосторонних последовательных протоколов.

Заключение

Подпороговые каналы, по существу, являются стеганографическими каналами. Такие криптографические конструкции, как схемы электронной подписи или аутентификации пользователя позволяют строить стеганографические каналы, существование которых, сделав предположение о криптографической стойкости таких схем, невозможно обнаружить в принципе.

Тем не менее разработанные методы могут с успехом применяться для защиты схем интерактивной идентификации удаленного абонента системы, а также схем электронной подписи от создания (но не обнаружения) в них подпороговых каналов. Кроме того, данный метод защиты от формирования широкополосных подпороговых каналов в схемах интерактивной идентификации может применяться и в других конструкциях, например, в различных интерактивных протоколах аутентификации сообщений и ключевого обмена, где в качестве параметров протокола используются некоторые случайные последовательности.

Примечания

1 См.: *Simmons G.J.* The Prisoners' Problem and Subliminal Channel // Proceedings of International Conference on Advances in Cryptology – CRYPTO'83. N. Y.: Plenum Press, 1984. P. 51–67; *Idem.* The Subliminal Channel and Digital Signatures // Lecture Notes in Computer Science – Eurocrypt'84. 1984. Vol. 209. P. 364–378.

2 См.: *Казарин О.В., Курило А.П., Ухлинов Л.М.* Метод защиты от создания подпороговых каналов в телекоммуникационных системах // Сборник докладов Международной конференции «Нейронные технологии обработки информации» IP+NN'96. 1996. С. 98–99.

3 См.: *Казарин О.В., Ухлинов Л.М.* Интерактивная система доказательств для интеллектуальных средств контроля доступа к информационно-вычислительным ресурсам // Автоматика и телемеханика. 1993. № 11. С. 167–175; *Казарин О.В.* Эффективные схемы интерактивной идентификации для систем распознавания «свой – чужой» // Вопросы защиты информации. 1995. № 3 (30). С. 54–57.

4 См.: *Казарин О.В.* Конвертируемые и селективно конвертируемые схемы подписи с верификацией по запросу // Автоматика и телемеханика. 1998. № 6. С. 178–188.

5 См.: *Варновский Н.П.* Криптография и теория сложности // Введение в криптографию / Под. общ. ред. В.В. Ященко. М.: МЦНМО, 2012. С. 27–43; *Казарин О.В.* Методология защиты программного обеспечения. М.: МЦНМО, 2009. 464 с.

6 См.: *Казарин О.В.* Эффективные схемы интерактивной идентификации...

7 См.: *Canetti R.* Universally Composable Security: a New Paradigm for Cryptographic Protocols // Lecture Notes in Computer Science. 42nd Foundation of Computer Sciences Conference. 2001. P. 136–145.

Д.А. Иванов, А.П. Никитин

ПРОТИВОДЕЙСТВИЕ АНАЛИЗУ КЛАВИАТУРНОГО ПОЧЕРКА

В данной работе авторами предлагается методика защиты пользователя от идентификации его по клавиатурному почерку. Рассмотрены как программные, так и аппаратные варианты реализации модуля защиты от анализа клавиатурного почерка. Предложены две принципиальные схемы реализации модуля, размещаемого между клавиатурой и компьютером. Также приведена принципиальная схема построения аппаратного модуля для порта PS/2.

Ключевые слова: клавиатурный почерк, идентификация пользователя, анонимизация.

В силу высокой индивидуальности биометрических параметров человека возможно построение систем идентификации пользователей на их основе. Широкое распространение таких методов делает необходимым рассмотрение также методов защиты от идентификации такого рода.

Согласно действующему законодательству Российской Федерации и ряда других стран, человек имеет право на анонимность своих действий в сети интернета[1].

В отличие от пароля или токена, биометрические данные не могут быть переданы другому лицу. Применение же методов биометрической идентификации позволяет с большой степенью достоверности идентифицировать именно человека, а не учетную запись или компьютер, на котором работает пользователь. Например, существуют методики, позволяющие идентифицировать человека по его клавиатурному почерку, с вероятностью порядка 0,91[2].

Также использование идентификации по клавиатурному почерку может проводиться незаметно для пользователя, в том числе

и при вводе текста через веб-браузер. Таким образом, недобросовестный владелец веб-сайта может собирать информацию о пользователе и в дальнейшем использовать ее в своих целях.

Основным параметром, который используется при анализе клавиатурного почерка, является время между событиями клавиатуры (т. е. нажатием кнопки и ее отпусканием). Идентификация пользователя строится на анализе длительности интервалов, соответствующих каждой комбинации клавиш (рис. 1.)

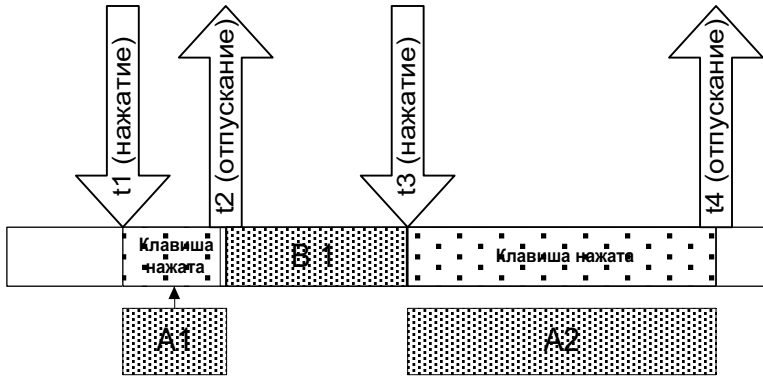


Рис. 1. Временная диаграмма клавиатурного почерка

Очевидно, что наиболее действенным методом анонимизации является выравнивание времени между событиями клавиатуры. В таком случае, идентификация конкретного пользователя станет теоретически невозможной. На рис. 2 приведена временная диаграмма анонимизированного почерка.

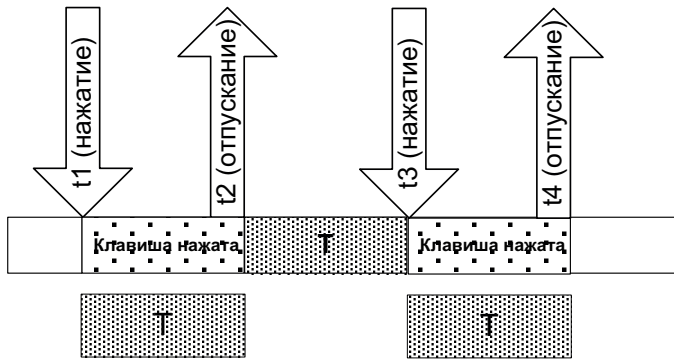


Рис. 2. Временная диаграмма анонимизированного почерка

Для реализации схемы изменения клавиатурного почерка предлагается использование программного или аппаратного модуля. Возможны следующие основные места встраивания такого модуля:

- непосредственно в клавиатуру (программно-аппаратный или аппаратный модуль);
- непосредственно в обработчик событий клавиатуры операционной системы (программный модуль);
- между клавиатурой и компьютером (аппаратный модуль в разрыв кабеля).

Рассмотрим каждый из вариантов более подробно.

Встраивание модуля непосредственно в клавиатуру требует внесения изменений в конструкцию клавиатуры и, возможно, в ее драйвер[3]. Каких-либо принципиальных преимуществ, кроме визуальной скрытности, такое размещение модуля не несет.

Программный модуль обработки событий клавиатуры не требует каких-либо аппаратных компонент, однако его функционирование зависит от операционной системы. По сути, каждая операционная система требует своего программного модуля. Такой модуль не зависит от того, через какой порт подключается клавиатура.

Установка модуля изменения клавиатурного почерка в разрыв кабеля позволяет использовать его на любой ОС. Недостатком же такого решения является зависимость от порта подключения (наиболее часто используются PS/2 или USB).

В данной работе рассмотрим такую схему установки модуля защиты от анализа клавиатурного почерка в разрыв кабеля.

Возможны следующие алгоритмы работы такого модуля.

Для реализации схемы предлагается использование буфера, где регистрируются события клавиатуры, а по окончании ввода текста содержимое буфера отправляется на компьютер. Для защиты от переполнения буфера предлагается использовать два буфера одинакового размера. С начала работы идет сохранение событий клавиатуры в один буфер. При его переполнении начинается сохранение во второй, а содержимое передается на компьютер. При передаче информации на компьютер интервалы между событиями выравниваются. Таким образом, достигается полная анонимность почерка, так как устраняются любые индивидуальные особенности почерка.

Вторая схема построения модуля предполагает наличие некоего временного интервала T, длина которого заведомо превышает длину интервала при печати. Сообщения о событиях клавиатуры сохраняются в памяти и отправляются с фиксированными интервалами T. В случае, если интервал между двумя событиями больше

чем T, отправка следующего сообщения о событии происходит в ближайший после сообщения интервал T.

Рассмотрим построение модуля, реализующего обе схемы и работающего с клавиатурами, которые подключаются через порт PS/2. В состав модуля входят аппаратные компоненты:

- микроконтроллер Atmel AT89C2051[4];
- память EEPROM типа AT24C512[5];
- кварцевый генератор 12 MHz;
- два конденсатора 33p;
- конденсатор 10 uF;
- резистор 10 k.

После включения питания модуль защиты от снятия клавиатурного почерка (МЗКП) по умолчанию включается в первый режим. В этом режиме функционирует первая схема, описанная выше. В этом режиме работы предусмотрено наличие двух буферов по 255 кбайт. При заполнении активного (первого) буфера более чем на 230 кбайт происходит переключение записи событий клавиатуры во второй буфер. Далее содержимое первого буфера автоматически пересылается на компьютер, и он становится резервным. Пересылка содержимого активного буфера на компьютер происходит при нажатии кнопки. При пересылке используются равные интервалы между событиями клавиатуры. Интервалы выбраны наименьшей возможной длины с целью уменьшить время передачи.

Также при нажатии на кнопку происходит переключение в другой режим работы, реализующий вторую схему (описана выше). Экспериментально было установлено, что необходимо наличие двух фиксированных интервалов, один из которых соответствует времени удержания клавиши клавиатуры в нажатом состоянии, а второй – интервалу между нажатием клавиш. В качестве базовых были выбраны интервалы, приведенные в таблице.

Таблица 1

Средние значения времени удержания клавиш и интервалов между нажатиями двух клавиш

Тип клавиатуры	Время удержания клавиши, мс	Время между нажатием клавиш, мс
С коротким ходом клавиш (клавиатура ноутбука)	100	150
С длинным ходом клавиш (стандартная клавиатура)	150	200

Для экспериментов по замерам времени применялись следующие клавиатуры:

- с интерфейсом PS/2:
  - A4TechKLS-7MU и GeniusSlimStar 320 – с коротким ходом клавиш;
  - Mitsumikfk-ea4saиDefenderkb 21e – с длинным ходом клавиш;
- с интерфейсом USB[6]:
  - LogitechK230 и DefenderGalileo 4920 – с коротким ходом клавиш;
  - LogitechG510s и GeniusGK-1000 – с длинным ходом клавиш.

Принципиальная схема МЗКП приведена на рис. 3

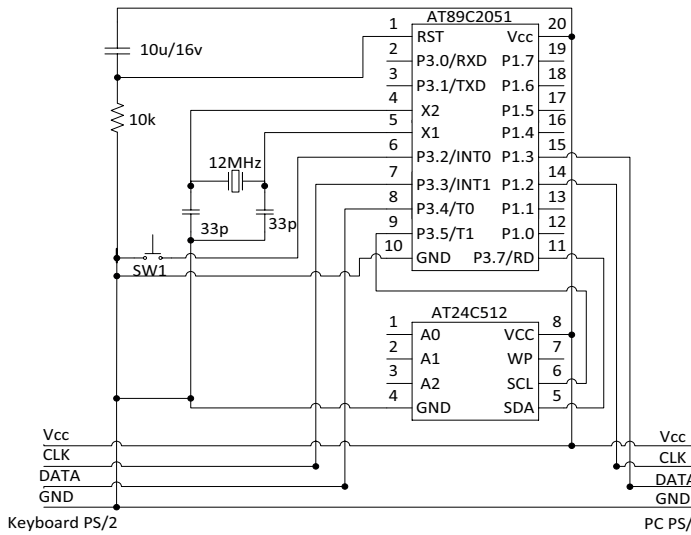


Рис. 3. Принципиальная схема модуля защиты

Проведенные авторами исследования показывают, что предложенный модуль идентификации позволяет устранить практически полностью индивидуальные особенности почерка при работе в первом режиме. Основным недостатком данного режима является затрудненная для пользователя работа при наборе текстов существенной длины. Выходом может служить следующая схема рабо-

ты пользователя: набор текста в первом режиме, отправка его на компьютер и внесение в него правки во втором режиме.

При работе во втором режиме можно установить различные интервалы, соответствующие скорости печати каждого конкретного пользователя. Эта же особенность является и недостатком, так как требуется подстройка под каждого конкретного пользователя. Опция автоматической подстройки в текущем прототипе не реализована. Возможна только ручная установка интервалов.

Заключение

В данной работе авторами рассмотрен ряд возможных методик защиты пользователя от идентификации его по клавиатурному почерку. Приведены возможные способы реализации модуля защиты. Сформулированы преимущества и недостатки как программных, так и аппаратных вариантов реализации.

Предложены две принципиальные схемы реализации модуля, размещаемого между клавиатурой и компьютером. Также приведена принципиальная схема построения аппаратного модуля для порта PS/2.

Примечания

[1] См.: Конституция РФ. Гл. 2. Ст. 23–24 [Электронный ресурс] // Президент России. URL: http://constitution.kremlin.ru (дата обращения: 22.04.2014).

[2] См.: *Григорьев В.Р., Никитин А.П.* Использование статических методов для биометрической идентификации пользователя // Вестник РГГУ. Сер. «Информатика. Защита информации. Математика». 2012. № 14. С. 135–142.

[3] Такие изменения требуются в том случае, когда необходимо изменить систему передачи сообщений клавиатуры.

[4] См.: Datasheet AT89C2051 [Электронный ресурс] // Сайт компании Atmel. URL: http://www.atmel.com/images/doc0368.pdf (дата обращения: 22.04.2014).

[5] См.: Datasheet AT24C512 [Электронный ресурс] // Сайт компании Atmel. URL: http://www.atmel.com/Images/doc1116.pdf (дата обращения: 22.04.2014).

[6] В ходе исследований было установлено, что характерное время между событиями клавиатуры зависит от типа самой клавиатуры, а не от интерфейса подключения.

С.В. Запечников

ВИЗУАЛИЗАЦИЯ ДАННЫХ И ПРОЦЕССОВ С ИСПОЛЬЗОВАНИЕМ КРОССПЛАТФОРМЕННОГО ПРОГРАММНОГО ИНТЕРФЕЙСА OPENGL*

Статья посвящена анализу возможностей визуализации данных и процессов, представимых в виде трехмерных объектов, комплексов объектов (сцен) или трехмерной анимации, с применением кроссплатформенных программных библиотек, поддерживающих интерфейс OpenGL. Выявляется существо математических и технических задач, которые позволяют решать эти программные средства, в частности, задач геометрического задания трехмерных объектов, проецирования их на экранную плоскость, управления видеопамятью при отображении графических объектов и др. Выделенные задачи систематизируются, для каждой из них по возможности приводится формальная постановка задачи и указываются способы решения. Для большинства обсуждаемых задач формулируются практические рекомендации по рациональному использованию возможностей интерфейса OpenGL при создании прикладных программ, требующих трехмерной визуализации.

Ключевые слова: визуализация данных, трехмерная графика, геометрические преобразования, интерфейс прикладного программирования, кроссплатформенное программное обеспечение, спецификация OpenGL.

В настоящее время отображение трехмерных объектов, сцен и анимаций является составной частью многих информационных технологий: приложений для визуализации результатов научных экспериментов, численного и имитационного моделирования, систем автоматизированного проектирования, геоинформационных систем, мультимедийных приложений и многих других видов

* Автор выражает благодарность В.Ю. Ефимову за предоставленные материалы по интерфейсу OpenGL и примеры исходных текстов программ, написанных с использованием интерфейса OpenGL.

программного обеспечения. Научное и практическое значение этой области знаний существенно повышается в связи с разрастанием круга задач, связанных с обработкой так называемых больших данных (big data) – сверхбольших массивов слабоструктурированных данных, обработка которых должна осуществляться в реальном масштабе времени. Реализация функциональности, позволяющей отображать в реальном масштабе времени сложные трехмерных графические объекты и сцены на экране компьютеров, является довольно сложной задачей и требует, как правило, совместного применения специализированной аппаратуры (в частности, высокопроизводительных видеокарт) и создаваемого для таких систем специального программного обеспечения.

Один из самых широко распространенных инструментариев для разработки программного обеспечения систем визуализации трехмерных объектов и сцен основан на спецификации интерфейса прикладного программирования OpenGL (Open Graphics Library). В своей основе эта спецификация определяет независимый от языка программирования и аппаратно-программной платформы кроссплатформенный, программный интерфейс для создания прикладных программ, использующих двухмерную и трехмерную компьютерную графику. Спецификация OpenGL включает более 250 функций для рисования сложных трехмерных сцен из простых примитивов. Текущая версия спецификации имеет обозначение OpenGL 4.4. Спецификация продолжает активно развиваться под эгидой консорциума ARB (Architecture Review Board).

Производители аппаратного обеспечения создают свои реализации библиотек, поддерживающих общую спецификация OpenGL. Таким образом, с точки зрения практики реализации систем визуализации трехмерных объектов, OpenGL – это в каждом конкретном случае некоторый набор библиотек, поддерживающий определенный круг аппаратных платформ, операционных систем и языков программирования. Реализации библиотек, как правило, имеют привязку к большинству современных языков программирования, таких как Java, C++, C#, Python, Perl и др.

Существующая литература, посвященная моделированию и визуализации трехмерных объектов, достаточно четко делится на две категории: одна из них – монографии и учебные пособия по теоретическим вопросам компьютерной графики[1], другая – практические пособия для системных архитекторов и программистов[2]. В то же время практически отсутствует литература, в которой выделялись бы типовые научно-практические задачи, возникающие при реализации систем визуализации трехмерных объектов, и об-

суждались бы сценарии и особенности их реализации при помощи широко доступного программного инструментария.

В настоящей статье предпринимается попытка проанализировать ядро спецификации OpenGL, по возможности максимально абстрагируясь от конкретных особенностей ее реализации для разных платформ и языков программирования. Однако ради определенности вся нотация и примеры функций приводятся в формате, соответствующем синтаксису языка программирования C++. Анализ проводится в первую очередь с целью выявления существа тех математических (главным образом, геометрических) и технических задач, которые позволяет решать эта библиотека. Выделенные таким образом задачи систематизируются, для каждой из них по возможности приводится формальная постановка задачи. Кроме того, для каждой из обсуждаемых в статье задач автор попытался сформулировать практические рекомендации по рациональному использованию возможностей OpenGL при создании приложений, требующих визуализации трехмерных объектов.

Общие сведения об интерфейсе OpenGL

Интерфейс прикладного программирования (API – Application Programming Interface) OpenGL позволяет стандартным образом использовать при разработке программного обеспечения возможности отображения трехмерной графики, реализуемые современными компьютерными видеокартами, при необходимости их эмулируя[3]. При этом снимается нагрузка с центрального процессора, так как вычисления графики переносятся на видеокарту, которая, вообще говоря, лучше для этого подходит по своей производительности. Иными словами, используется модель вычислений «клиент–сервер», где клиентом является программа, работающая на центральном процессоре, а сервером – программа, работающая на процессоре или процессорах видеокарты. При этом клиентом используются вызовы специальных функций, чтобы взаимодействовать с сервером. Если видеокарта не поддерживает некоторую команду, то эта команда будет эмулирована на центральном процессоре, если это возможно.

Вычисления с использованием ресурсов видеокарт в настоящее время получили широкое распространение и в других приложениях, не связанных с компьютерной графикой. Ярким примером может служить эмиссия криптовалют, таких как Bitcoin, с использованием ферм видеокарт.

OpenGL реализует конвейер обработки графики, работающий следующим образом.

1. Клиент вызовами функций API формирует в буфере на своей стороне последовательность команд и данных.

2. Автоматически при заполнении буфера либо по вызову функции glFlush() или glFinish() происходит отправка команд и данных в виде пакета на видеокарту. Отличие этих функций заключается в том, что glFinish() не вернет управление, пока пакет не пройдет все последующие этапы конвейера.

3. Видеокарта выполняет заданные команды над полученными данными, такие как позиционирование точек (преобразование их координат), проецирование, наложение цветов и/или текстур в соответствии с настройками света, фильтрация, постобработка графики и др. Геометрические преобразования имеют следующую очередность.

3.1. Преобразование наблюдения модели: к вершинам заданных примитивов применяются геометрические преобразования поворота и сдвига, выраженные в матрице наблюдения модели.

3.2. Преобразование проецирования: трехмерные координаты вершин преобразовываются в нормализованные экранные координаты (НЭК); данное преобразование выражено в матрице проецирования и операции перспективного деления.

3.3. Преобразование области наблюдения: НЭК вершин преобразуются в двухмерные экранные координаты с учетом глубины и формируют изображение.

4. Результат выводится в один из буферов: буфер кадров, буфер выбора и буфер обратной связи. При выводе в буфер кадров формируется растровое изображение, которое, в частности, выводится на экран. В буфер выбора выводятся определенные клиентом данные об объектах, которые могли бы быть изображены при выводе в буфер кадров. В буфер обратной связи выводятся данные о результате преобразования объектов.

При вызове функций интерфейса OpenGL из программ, написанных на языке C++, должны использоваться следующие основные заголовочные файлы[4]:

- gl.h – файл, который содержит все определения интерфейса;
- glu.h – файл, который содержит определения библиотеки OpenGL Utility Library (GLU), реализующей некоторые дополнительные возможности интерфейса;
- glut.h – файл, который содержит определения библиотеки The OpenGL Utility Toolkit (GLUT), реализующей кроссплатформенный интерфейс программирования оконных приложений, использующих OpenGL.

Задание трехмерных объектов в OpenGL

Рассмотрим базовые задачи трехмерной графики, которые лежат в основе практически всех методов визуализации трехмерных объектов.

2.1. Примитивы и сложные объекты

Одним из элементарных объектов в трехмерной графике является вершина (в терминологии OpenGL – vertex), представленная трехмерным вектором $\bar{p} = (x, y, z)$ с масштабным коэффициентом w (смысл масштабного коэффициента будет рассмотрен ниже). В терминах языка C++:

typedef struct{GLfloat x; GLfloat y; GLfloat z; GLfloat w;}VECTOR4F,

или, что эквивалентно:

typedef struct{GLfloat f[4];}VECTOR4F.

Вершину можно интерпретировать как точку (GL_POINT). Из одной вершины в другую можно провести отрезок (GL_LINES), через упорядоченную последовательность вершин можно провести ломаную (GL_LINE_STRIP), а также замкнутую ломаную (GL_LINE_LOOP), для которой автоматически достраивается отрезок из последней вершины в первую. Из трех вершин можно получить треугольник (GL_TRIANGLES). Из четырех и более вершин (считая, что точки добавляются последовательно одна за другой) можно получить ленту треугольников (GL_TRIANGLE_STRIP), в которой каждый следующий треугольник строится на последних трех добавленных вершинах, и веер треугольников (GL_TRIANGLE_FAN), в котором каждый следующий треугольник строится на последних двух вершинах и первой.

Существуют еще такие примитивы, как четырехугольники (GL_QUADS) и многоугольники (GL_POLYGON), для задания которых нужно использовать соответственно четыре или любое число вершин, большее трех. Возможно построить ленту четырехугольников (GL_QUAD_STRIP), где каждый следующий четырехугольник строится на последних двух вершинах предыдущего и на новых двух. При этом вершины должны лежать в одной плоскости, а сами многоугольники должны быть выпуклыми. Заметим, что все библиотеки OpenGL оптимизированы под быструю обработку выпуклых многоугольников.

Чтобы отобразить на экране что-то из вышеперечисленного, требуется вызвать функцию

void glBegin(GLenum mode),

где в качестве mode указывают отображаемый объект: GL_POINTS, GL_TRIANGLES и др. Далее необходимо перечислить требуемое количество вершин объекта при помощи вызова функции:

void glVertex4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)

или

void glVertex4fv(GLfloat *p4),

где p4 – указатель на массив как минимум из четырех точек GLfloat, например f в структуре VECTOR4F. Однако чаще всего используются функции

void glVertex3f(GLfloat x, GLfloat y, GLfloat z),

void glVertex3fv(GLfloat *p3),

где p3 – указатель на массив как минимум из трех GLfloat (f из VECTOR4F тоже соответствует этому условию), при этом по умолчанию в OpenGL считается w = 1. В конце геометрического построения необходимо вызвать функцию

void glEnd().

Более сложные объекты и геометрические фигуры определяются через перечисленные выше примитивы. Их построение сводится к комбинированию примитивов.

OpenGL имеет только набор перечисленных геометрических примитивов, из которых создаются все трехмерные объекты. Подобный уровень детализации не всегда бывает удобен при создании сложной графики. Поэтому поверх OpenGL были созданы высокоуровневые библиотеки, такие как Open Inventor, VTK, GLU, GLEW, SDL, GLM и др., позволяющие оперировать более сложными трехмерными объектами. В частности, широко применяемая библиотека GLU, скрывая от программиста многие математические тонкости построения, дает возможность быстро запрограммировать некоторые аналитически задаваемые поверхности.

2.2. Понятие направления обхода многоугольников

Порядок обхода вершин при построении треугольника (в общем случае многоугольника) позволяет задать его переднюю и заднюю сторону. Если вершины треугольника обходятся по часовой стрелке, то видимой для зрителя считается задняя сторона треугольника. По умолчанию используется именно это правило, однако его возможно переопределить вызовом функции glFrontFace(GLenum mode): макроопределение GL_CCW, переданное ей в качестве параметра, определяет описанное выше определение передней и задней сторон, а GL_CW – противоположное.

В математических терминах обход вершин треугольника можно проиллюстрировать следующим образом. Пусть заданы вершины

p_1, p_2, p_3. Построим на них два вектора $\vec{v_1} = p_1 - p_2$ и $\vec{v_2} = p_3 - p_1$. Тогда их нормализованное векторное произведение будет вектором-нормалью треугольника с координатами *a, b, c*.

$$\vec{n} = \frac{[\vec{v_1}, \vec{v_2}]}{|[\vec{v_1}, \vec{v_2}]|} = (a, b, c) \qquad (1)$$

Если провести через указанные точки плоскость *D*, то $\vec{n}$ будет нормалью к ней. Нормаль будет указывать от плоскости в сторону положительного полупространства, из которого будет видна передняя сторона треугольника. Можно получить нормальное уравнение плоскости в пространстве $D(x, y, z) = ax + by + cz + d = 0$, которому удовлетворяют все точки (x, y, z), принадлежащие *D*. Коэффициент *d* можно выразить, подставив вместо *x, y, z* координаты любой из точек плоскости *D*, например (p_1, p_2, p_3). Если точка $p' = (x', y', z')$ не принадлежит *D*, то $|D(x', y', z')|$ будет кратчайшим расстоянием от точки p' до плоскости *D*. В зависимости от того, с какой стороны от плоскости расположена p', значения выражения $D(x', y', z')$ будут иметь разные знаки, откуда и возникают понятия положительного и отрицательного полупространств.

Когда определена передняя и задняя стороны многоугольников, можно настроить интерполяцию точек при помощи функции

void glPolygonMode(GLenum face, GLenum mode),

где face указывает, что настройка относится к передней (GL_FRONT), задней (GL_BACK) или к обеим сторонам (GL_FRONT_AND_BACK), а mode определяет, что интерполироваться будут все точки (GL_FILL) (например, по заданным трем вершинам треугольника будут построены его внутренние точки, чтобы он казался сплошным), только контуры (GL_LINE) или только вершины (GL_POINT). При интерполяции только контуров (вершин) можно указать, какие из них отображать не следует. Это делается при помощи вызова функции glEdgeFlag(false) перед группой вызовов glVertex*, но затем нужно явно указать, какие контуры (вершины) следует рисовать, вызвав функцию glEdgeFlag(true). Возможно также полностью запретить внутреннюю интерполяцию многоугольников с заданной стороны при помощи вызова функции

void glCullFace(GLenum mode),

запретив при этом интерполяцию вызовом функции glEnable(GL_CULL_FACE), где mode определяет неинтерполируемую сторону (GL_FRONT, GL_BACK), а при указании GL_FRONT_AND_BACK внутренняя часть многоугольников вообще не будет интерполироваться. Запрет внутренней интерполяции не влияет на интерполяцию контуров и вершин.

2.3. Позиционирование и ориентация объектов

Все рассмотренные выше фигуры и примитивы строятся относительно точки отсчета (0; 0; 0). Однако почти всегда требуется одновременно рисовать несколько объектов, каждый из которых задан относительно этой же точки. При этом требуется, чтобы объекты могли перемещаться. Тогда целесообразно ввести некоторую глобальную систему координат с центром в точке (0; 0; 0), а точкам отсчета всех объектов приписываются координаты в этой глобальной системе. Кроме того, часто возникает потребность поворота объекта. Иными словами, позиционирование объекта сводится к преобразованиям параллельного переноса и поворота вокруг точки (0; 0; 0).

Такие преобразования удобнее всего выражаются в терминах матричного исчисления. В OpenGL реализован именно такой подход. Для позиционирования объекта используется матрица наблюдения модели размером 4 × 4:

$$\begin{pmatrix} x_{в_1}, & x_{в_2}, & x_{в_3}, & x_t \\ y_{в_1}, & y_{в_2}, & y_{в_3}, & y_t \\ z_{в_1}, & z_{в_2}, & z_{в_3}, & z_t \\ 0, & 0, & 0, & 1 \end{pmatrix}.$$

Левая верхняя подматрица размером 3 × 3 характеризует поворот точек объекта относительно его же точки отсчета. Ее можно описать следующим образом. Известно, что точки модели заданы координатами относительно осей *x, y, z*, или, что то же самое, координаты $(x_p;\ y_p;\ z_p)$ точки *P* есть разложение вектора $\vec{p}$, проведенного из точки (0; 0; 0) в данную точку $P = (x_p; y_p; z_p)$ в базисе $\vec{i} = (1; 0; 0)$, $\vec{j} = (0; 1; 0)$, $\vec{k} = (0; 0; 1)$. Иными словами,

$$\vec{p} = x_p \cdot \vec{i} + y_p \cdot \vec{j} + z_p \cdot \vec{k}, \qquad (2)$$

где вектора $\vec{i},\ \vec{j},\ \vec{k}$ образуют ортонормированный базис (ОНБ) на осях *x, y, z* соответственно. Таким образом, все точки модели

жестко привязаны к осям, и поворот объекта есть поворот этих осей. Для этой цели несколько изменим обозначения. Пусть вектора $\vec{i},\ \vec{j},\ \vec{k}$ будут характеризовать *только* оси глобальной системы отсчета. Оси же произвольного объекта будем характеризовать векторами $\vec{e}_1, \vec{e}_2, \vec{e}_3$ (требуется, чтобы они тоже всегда составляли ОНБ). Относительно точек объекта эти векторы считаются ортами осей: $\vec{e}_1 = \{1;\ 0;\ 0\}$, $\vec{e}_2 = \{0;\ 1;\ 0\}$, $\vec{e}_3 = \{0;\ 0;\ 1\}$. Предположим, что внутренняя система координат была повернута. Тогда числа $(x_{в_1},\ y_{в_1},\ z_{в_1})$ являются новыми координатами вектора $\vec{e}_1$ (другим вектором $\vec{e}'_1$) после поворота, т.е. разложением в базисе $[\vec{i},\ \vec{j},\ \vec{k}]$, а числа $(x_{в_2},\ y_{в_2},\ z_{в_2})$ и $(x_{в_3},\ y_{в_3},\ z_{в_3})$ – новыми координатами векторов $\vec{e}_2$ и $\vec{e}_3$ соответственно. Поворот объекта есть переход от одного ОНБ $\{\vec{e}_1, \vec{e}_2, \vec{e}_3\}$ к другому $\{\vec{e}'_1, \vec{e}'_2, \vec{e}'_3\}$. Теперь положим, что точка P задана координатами $(x_p,\ y_p,\ z_p)$ в повернутом базисе $\{\vec{e}'_1, \vec{e}'_2, \vec{e}'_3\}$, т. е. ее координаты есть координаты вектора p:

$$\vec{p} = x_p \cdot \vec{e}'_1 + y_p \cdot \vec{e}'_2 + z_p \cdot \vec{e}'_3 = (\vec{e}'_1, \vec{e}'_2, \vec{e}'_3) \cdot \begin{pmatrix} x_p \\ y_p \\ z_p \end{pmatrix}$$

(это выражение есть обобщение выражения (2)). В свою очередь, вектор $\vec{e}'_i = \{x_{в_i};\ y_{в_i};\ z_{в_i}\}$ задан относительно базиса $\{\vec{e}_1, \vec{e}_2, \vec{e}_3\}$, т.е.:

$$\vec{e}'_i = x_{в_i} \cdot \vec{e}_1 + y_{в_i} \cdot \vec{e}_2 + z_{в_i} \cdot \vec{e}_3 = (\vec{e}_1, \vec{e}_2, \vec{e}_3) \cdot \begin{pmatrix} x_{в_i} \\ y_{в_i} \\ z_{в_i} \end{pmatrix}$$

где $i \in \{1, 2, 3\}$.

В матричном виде:

$$\vec{p} = \left((\vec{e}_1; \vec{e}_2; \vec{e}_3) \cdot \begin{pmatrix} x_{в_1} \\ y_{в_1} \\ z_{в_1} \end{pmatrix}; (\vec{e}_1; \vec{e}_2; \vec{e}_3) \cdot \begin{pmatrix} x_{в_2} \\ y_{в_2} \\ z_{в_2} \end{pmatrix}; (\vec{e}_1; \vec{e}_2; \vec{e}_3) \cdot \begin{pmatrix} x_{в_3} \\ y_{в_3} \\ z_{в_3} \end{pmatrix} \right) \cdot \begin{pmatrix} x_p \\ y_p \\ z_p \end{pmatrix} =$$

$$= (\vec{e_1}; \vec{e_2}; \vec{e_3}) \cdot \begin{pmatrix} x_{в_1}, ; x_{в_2}, ; x_{в_3}, \\ y_{в_1}, ; y_{в_2}, ; y_{в_3}, \\ z_{в_1}, ; z_{в_2}, ; z_{в_3}, \end{pmatrix} \cdot \begin{pmatrix} x_p \\ y_p \\ z_p \end{pmatrix}.$$

Отсюда следует, что:

$$\{\vec{e'_1}, \vec{e'_2}, \vec{e'_3}\} = (\vec{e_1}; \vec{e_2}; \vec{e_3}) \cdot \begin{pmatrix} x_{в_1}, ; x_{в_2}, ; x_{в_3}, \\ y_{в_1}, ; y_{в_2}, ; y_{в_3}, \\ z_{в_1}, ; z_{в_2}, ; z_{в_3}, \end{pmatrix},$$

где $\begin{pmatrix} x_{в_1}, ; x_{в_2}, ; x_{в_3}, \\ y_{в_1}, ; y_{в_2}, ; y_{в_3}, \\ z_{в_1}, ; z_{в_2}, ; z_{в_3}, \end{pmatrix}$ – матрица перехода от базиса $\{\vec{e_1}, \vec{e_2}, \vec{e_3}\}$ к базису $\{\vec{e'_1}, \vec{e'_2}, \vec{e'_3}\}$, где в столбцах содержатся координаты нового базиса в старом (базисы не обязательно должны быть ортонормированными). В общем случае, если $\{\vec{e_1}, \vec{e_2}, \vec{e_3}\} \neq \{\vec{i}, \vec{j}, \vec{k}\}$, то существует матрица перехода T, такая что $\{\vec{e_1}, \vec{e_2}, \vec{e_3}\} = \{\vec{i}, \vec{j}, \vec{k}\}$, т.е. преобразование поворота может быть представлено произведением нескольких матриц. Левая верхняя подматрица размером 3×3 матрицы наблюдения и есть матрица перехода от $\{\vec{i}, \vec{j}, \vec{k}\}$ к $\{\vec{e_1}, \vec{e_2}, \vec{e_3}\}$, где $\{\vec{e_1}, \vec{e_2}, \vec{e_3}\} \neq \{\vec{i}, \vec{j}, \vec{k}\}$ в общем случае, а $\vec{e_1}, \vec{e_2}, \vec{e_3}$ есть орты, отложенные по повернутым осям x, y, z объекта.

Правый верхний вектор-столбец (подматрица размером 3×1) матрицы наблюдения объекта характеризует смещение внутренней точки отсчета объекта относительно глобальной точки отсчета.

Для вычисления положения каждой точки модели $p_i = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}$ в глобальной системе координат следует выполнить следующее преобразование:

$$p_i' = \begin{pmatrix} x_i' \\ y_i' \\ z_i' \end{pmatrix} = \cdot \begin{pmatrix} x_{в_1}, x_{в_2}, x_{в_3} \\ y_{в_1}, y_{в_2}, y_{в_3} \\ z_{в_1}, z_{в_2}, z_{в_3} \end{pmatrix} \cdot \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} + \begin{pmatrix} x_t \\ y_t \\ z_t \end{pmatrix}.$$

Однако существует упомянутый выше масштабный коэффициент w_i, с учетом которого данное положение точки вычисляется так:

$$p_i' = \begin{pmatrix} x_i' \\ y_i' \\ z_i' \end{pmatrix} = \cdot \begin{pmatrix} x_{в_1}, & x_{в_2}, & x_{в_3} \\ y_{в_1}, & y_{в_2}, & y_{в_3} \\ z_{в_1}, & z_{в_2}, & z_{в_3} \end{pmatrix} \cdot \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} + \begin{pmatrix} x_t \\ y_t \\ z_t \end{pmatrix} \cdot w_i .$$

Отсюда становится ясен смысл масштабного коэффициента: он определяет масштаб параллельного переноса координат точки в *глобальной* системе отсчета. В OpenGL это реализовано так:

$$\begin{pmatrix} x_i' \\ y_i' \\ z_i' \\ w_i \end{pmatrix} = \begin{pmatrix} x_{в_1}, & x_{в_2}, & x_{в_3}, & x_t \\ y_{в_1}, & y_{в_2}, & y_{в_3}, & y_t \\ z_{в_1}, & z_{в_2}, & z_{в_3}, & z_t \\ 0, & 0, & 0, & 1 \end{pmatrix} \cdot \begin{pmatrix} x_i \\ y_i \\ z_i \\ w_i \end{pmatrix} .$$

Нижняя строка, заданная таким образом, позволяет оставить масштабный коэффициент неизменным, но она так же позволяет изменять его в линейной зависимости от координат точки во внутренней системе отсчета. Манипуляции с этой строкой при визуализации трехмерных объектов применяются сравнительно редко, поэтому в настоящей работе они не рассматриваются.

Легко заметить, что единичная матрица

$$E = \begin{pmatrix} 1; 0; 0; 0 \\ 0; 1; 0; 0 \\ 0; 0; 1; 0 \\ 0; 0; 0; 1 \end{pmatrix}$$

описывает несмещенный и не повернутый объект, т. е. объект, чья система отсчета полностью совпадает с глобальной. Иными словами, *E* выражает тождественное преобразование положения объекта.

Алгоритм преобразования положения объекта реализован внутри OpenGL (т. е. не требуется явно перемножать матрицу на вектор, чтобы переместить/повернуть объект), а соответственная матрица носит название *матрицы наблюдения модели* (GL_MODELVIEW). Для хранения матрицы в памяти на стороне клиента (прикладной программы) можно использовать массив из 16 чисел с плавающей точкой, содержащий развертку по столбцам матрицы наблюдения модели:

typedef struct{ GLfloat a[16]; }MATRIX4X4.

При этом следует обращать внимание на принятую в OpenGL последовательность присвоения индексов элементам матрицы в массиве a [16]:

$$\begin{pmatrix} 0 & 4 & 8 & 12 \\ 1 & 5 & 9 & 13 \\ 2 & 6 & 10 & 14 \\ 3 & 7 & 11 & 15 \end{pmatrix}.$$

Эта развертка применима к любой матрице в OpenGL.

Таким образом, положение точки определяется в трехмерном пространстве ее координатами относительно некоторой точки отсчета и матрицей наблюдения модели.

2.4. Перемещение объектов

При визуализации процессов трехмерные объекты могут со временем изменять свое положение, в связи с чем возникает вопрос о получении новой матрицы наблюдения модели из предыдущей. Ранее было показано, что поворот объекта в пространстве выражается матрицей перехода от одного базиса к другому размером 3 × 3, но в OpenGL применяются матрицы 4 × 4. С этой целью в OpenGL существуют встроенные средства преобразования матриц, и, таким образом, программист может либо сам реализовать математику преобразований, либо воспользоваться готовыми функциями библиотеки, реализующей интерфейс OpenGL. Как правило, оба подхода используются совместно. Ниже будет рассмотрен ряд преобразований, а также их реализация средствами OpenGL.

Перенос объекта в глобальной системе координат из точки $A = (x_A, y_A, z_A)$ в точку $B = (x_B, y_B, z_B)$ выполняется прибавлением к текущей матрице наблюдения матрицы $T_{A \to B}$:

$$T_{A \to B} = \begin{pmatrix} 0; 0; 0; dx \\ 0; 0; 0; dy \\ 0; 0; 0; dz \\ 0; 0; 0; 0 \end{pmatrix}$$

где $dx = x_B - x_A$; $dy = y_B - y_A$; $dz = z_B - z_A$; .

В OpenGL нет явного метода для выполнения такого преобразования. Более того, несмотря на то что математически подобные преобразования выражаются произведением матриц, в OpenGL это преобразование задано сложением матриц.

Перенос объекта во внутренней системе отсчета на вектор $\vec{d} = (dx, dy, dz)$ выполняется умножением справа матрицы наблюдения объекта на матрицу $T_{\vec{d}}$:

$$T_{\vec{d}} = \begin{pmatrix} 1; 0; 0; dx \\ 0; 1; 0; dy \\ 0; 0; 1; dz \\ 0; 0; 0; 1 \end{pmatrix}.$$

т. е. координаты каждой точки объекта в глобальной системе отсчета будут вычислены так, как будто ее перенесли на вектор $\vec{d}$ во внутренней системе отсчета. Аналогичного преобразования можно добиться явно, прибавив $\vec{d}$ к координатам каждой точки объекта.

Это преобразование реализуется функцией:

void glTranslatef(GLfloat dx, GLfloat dy, GLfloat dz).

Оно может быть выполнено более рационально, если перенос выполняется вдоль только одной из осей x, y, z :

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_{в_1} \\ y_{в_1} \\ z_{в_1} \end{pmatrix} \cdot dx, \qquad \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_{в_2} \\ y_{в_2} \\ z_{в_2} \end{pmatrix} \cdot dy, \qquad \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_{в_3} \\ y_{в_3} \\ z_{в_3} \end{pmatrix} \cdot dz.$$

Поворот объекта может быть выполнен при помощи матрицы T_{angle} :

$$T_{angle} = \begin{pmatrix} x_{в_1}, & x_{в_2}, & x_{в_3}, & 0 \\ y_{в_1}, & y_{в_2}, & y_{в_3}, & 0 \\ z_{в_1}, & z_{в_2}, & z_{в_3}, & 0 \\ 0, & 0, & 0, & 1 \end{pmatrix},$$

где левая верхняя подматрица размером 3×3 есть описанная выше матрица перехода от одного базиса к другому. Такая матрица получается произведением матриц поворота вокруг осей внутренней системы координат объекта. Поворот вокруг осей x, y, z на угол angle (значения угла считаются положительными против часовой стрелки, если смотреть в направлении роста оси) может быть выполнен при помощи умножения матрицы наблюдения на матрицу , $\begin{pmatrix} T & 0 \\ 0; 0; 0; 1 \end{pmatrix}$, где T является соответственно одной из матриц:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & cos(angle) & -sin(angle) \\ 0 & sin(angle) & cos(angle) \end{pmatrix},$$

$$\begin{pmatrix} cos(angle) & 0 & -sin(angle) \\ 0 & 1 & 0 \\ sin(angle) & 0 & cos(angle) \end{pmatrix}, \qquad (3)$$

$$\begin{pmatrix} cos(angle) & -sin(angle) & 0 \\ sin(angle) & cos(angle) & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

В OpenGL для этих целей применяется функция

void glRotatef(GLfloat angle,GLfloat x,GLfloat y,GLfloat z),

которая осуществляет поворот на угол angle (в градусах) вокруг оси, сонаправленной вектору (x, y, z) и проходящей через начало внутренней системы координат объекта против часовой стрелки, если смотреть в сторону вектора. Вектор должен иметь единичную длину, иначе OpenGL нормализует его самостоятельно. Такое преобразование выполняется умножением на матрицу:

$$\begin{bmatrix} x^2(1-c)+c & xy(1-c)-zs & xz(1-c)+ys & 0 \\ yx(1-c)+zs & y^2(1-c)+c & yz(1-c)-xs & 0 \\ xz(1-c)-ys & yz(1-c)+xs & z^2(1-c)+c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

где $c = cos(angle)$, $s = sin(angle)$. Это преобразование содержит большое количество арифметических операций, поэтому для поворотов вокруг одной из осей рационально использовать одну из матриц (3).

Матричная арифметика в OpenGL

Рассмотрим теперь, как преобразования наблюдаемых объектов могут быть реализованы на практике. Как правило, при визуализа-

ции сложных процессов или результатов научных экспериментов необходимо отобразить множество объектов, каждому из которых должна быть сопоставлена матрица наблюдения (в тривиальном случае – единичная). При этом если объекты могут перемещаться с течением времени, то их матрицы следует умножать на матрицы соответствующих преобразований: это можно делать явно, запрограммировав операции умножения матриц, а можно – средствами OpenGL. Основные операции, которые необходимо реализовать при первом подходе, были рассмотрены выше. Далее рассмотрим подход с использованием встроенных средств матричной арифметики OpenGL.

В OpenGL существует понятие *текущей матрицы* – это матрица, над которой выполняются все преобразования. Чтобы определить текущую матрицу, следует вызвать функцию:

void glMatrixMode(GLenum mode).

При этом, если осуществляется позиционирование объекта в пространстве, необходимо задать матрицу наблюдения модели (mode = GL_MODELVIEW). Нижеперечисленные функции работают с текущей матрицей:

чтобы загрузить единичную матрицу, используется функция

void glLoadIdentity(void);

чтобы загрузить матрицу (например, из MATRIX4X4), используется функция

void glLoadMatrixf(const GLfloat *a);

чтобы умножить текущую матрицу на матрицу, хранимую программой-клиентом (например, из MATRIX4X4) справа, используется функция

void glMultMatrixf(const GLfloat *a).

Если нужно загрузить матрицу из MATRIX4X4, предварительно ее транспонировав, с целью последующего умножения, используется функция glLoadTransposeMatrix (glMultTransposeMatrix).

В OpenGL существует стек матриц (для каждого типа матриц, такого как GL_MODELVIEW и др. – свой), куда можно помещать матрицы. Например, в стеке следует сохранить текущую матрицу в случае, если планируется использование OpenGL для умножения других матриц). Для этого используются соответственно функции извлечения и помещения матрицы в стек:

void glPopMatrix(void),

void glPushMatrix(void).

Наконец, чтобы получить матрицу обратно в массив, следует воспользоваться функцией семейства glGet:

void glGetFloatv(GLenum pname, GLfloat *a),

где pname необходимо установить в GL_MODELVIEW_MATRIX для случая, когда операции происходят именно над матрицей наблюдения модели, или в GL_PROJECTION_MATRIX, если требуется матрица проецирования.

Проецирование

Выше были рассмотрены способы задания и размещения объектов в трехмерном пространстве. Однако, чтобы иметь возможность изобразить объекты на двухмерном экране компьютера, необходимо выполнить проецирование. Как было сказано выше, преобразование проецирования выполняется средствами OpenGL, но для этого разработчику приложения необходимо задать *матрицу проекции* (GL_PROJECTION), предварительно сделав ее текущей.

Далее рассмотрим проецирование двух видов: параллельное и перспективное.

При параллельном проецировании важными понятиями являются:

- объект наблюдения – множество точек, которые необходимо отобразить;
- плоскость проецирования – некоторая плоскость в трехмерном пространстве, на которую необходимо отобразить объект наблюдения.

Тогда *параллельное проецирование* – процесс нахождения точек пересечения плоскости проецирования и прямых, проходящих через точки объекта наблюдения параллельно нормали плоскости проецирования.

Перспективное проецирование основано на следующих понятиях:

- объект наблюдения;
- глаз наблюдателя – точка в трехмерном пространстве, откуда осуществляется обзор объекта или сцены;
- плоскость проецирования – некоторая плоскость в пространстве, не проходящая через глаз наблюдателя, на которую необходимо спроецировать объекты наблюдения.

Итак, *перспективное проецирование* – процесс нахождения точек пересечения плоскости наблюдения и прямых, проходящих через глаз наблюдателя и точки объектов наблюдения.

Фактически же в OpenGL любое проецирование осуществляется на плоскость, параллельную плоскости *xy*, и проходящей через точку (0, 0 – *nearVal*).

При любом виде проецирования координаты точек-проекций должны быть преобразованы в две координаты на плоскости проецирования.

4.1. Нормализованные экранные координаты и понятие глубины

Найти точки на плоскости проецирования недостаточно. Чтобы отобразить их на двухмерном экране, необходимо ввести на плоскости проецирования двухмерную систему координат. В качестве двухмерной системы координат на плоскости проецирования используются так называемые *нормализованные экранные координаты* (НЭК), значения которых в интервале [–1; 1] можно интерпретировать как находящиеся в пределах экрана, значение (0, 0) соответствует середине экрана. При отсчете НЭК принято, что направление оси x возрастает вправо, y – вверх. Вообще говоря, можно добиться того, что и точки с координатами, превышающими «экранный» интервал, можно будет видеть на экране. При этом на плоскости проецирования очерчивается так называемый прямоугольник наблюдения с углами в точках (*left, bottom, –nearVal*) и (*right, top, –nearVal*). Предположим, что *left* < *right, bottom* < *top*. Если это не так, то все равно без изменения формы прямоугольника можно поменять соответствующие координаты. Далее, если речь идет только о точках плоскости проецирования, будем опускать *z*-координату, равную *–nearVal*, так как она одинакова для всех точек этой плоскости. Если в ходе проецирования на плоскости проецирования была получена проекция (x_p, y_p), то, чтобы перевести ее в НЭК, следует определить ее координаты относительно середины прямоугольника наблюдения, а затем поделить на его ширину и высоту соответственно, т. е. нормализовать по прямоугольнику наблюдения. Следует также умножить результат на 2, чтобы нормализованные координаты имели требуемую величину 1 (или –1) на границе прямоугольника наблюдения:

$$(x_{p,n}, y_{p,n}) = 2 \cdot \frac{(x_p, y_p) - \left(\frac{right + left}{2}, \frac{top + bottom}{2}\right)}{(right - left, top - bottom)},$$

где под делением векторов подразумевается почленное деление соответствующих координат. Заметим, что прямоугольник наблюдения есть прообраз экрана, на котором наблюдается трехмерная картинка.

Часто бывает так, что точки, имеющие разные трехмерные координаты, перекрываются при проецировании (имеют одинаковые двухмерные координаты проекций). При этом их все равно следует различать в целях последующего определения приоритетов отрисовки на экране. Для этого вводится третья координата проецированной точки – глубина (ее геометрический смысл такой же, как и у координаты z).

Для учета глубины вместе с буфером кадров используется *z-буфер* (другими словами, *буфер глубины*), содержащий еще по одному числу для каждой точки буфера кадров – значение, характеризующее его близость к плоскости проецирования.

При проецировании точки всегда требуется вычислять значение ее z-координаты. Оно также должно лежать в интервале $[-1; 1]$, иначе точка не будет нарисована вовсе. Для этого z-координату точки требуется нормализовать. В этих целях вводится еще одна плоскость – плоскость отсечения, проходящая через точку $(0, 0 - farVal)$, $farVal > nearVal$ параллельно плоскости xy и, следовательно, параллельно плоскости проецирования. Значение $z = -1$ свидетельствует о том, что точка лежит в плоскости проецирования, при $z = 0$ точка находится строго между плоскостями, а при $z = 1$ – в плоскости отсечения.

Чтобы использовать z-буфер, необходимо определить критерий распределения приоритетов между перекрывающимися точками при помощи функции

void glDepthFunc(GLenum func),

причем значение ее аргумента func = GL_LESS соответствует естественному порядку: ближние точки рисуются поверх дальних. Полный перечень критериев можно найти в руководстве OpenGL 4 References Pages. Далее необходимо определить начальное заполнение буфера глубины вызовом функции glClearDepth(1), где 1 означает, что содержимое буфера кадров удалено на максимально возможное расстояние. Наконец, проверка глубины задействуется вызовом функции glEnable(GL_DEPTH_TEST). Все возможности, которые можно «разрешить» функцией glEnable, можно «запретить» функцией glDisable с тем же параметром, а чтобы узнать, «разрешена» ли та или иная возможность, следует вызвать функцию

GLboolean glIsEnabled(GLenum cap).

Перед очередным перерисовыванием содержимого экрана требуется вызывать функцию glClear(GL_DEPTH_BUFFER_BIT), чтобы загрузить в буфер глубины начальное заполнение.

Вообще говоря, процесс получения нормализованного значения зависит от способа проецирования: параллельного (ортогонального) или перспективного. Рассмотрим оба способа.

4.2. Параллельное проецирование

В случае параллельной проекции точки объекта наблюдения проецируются на плоскость наблюдения параллельно оси *z*. Координаты точек (x, y) и их проекции совпадают, остается только их нормализовать. При нормализации координаты z_p из нее следует вычесть координату по оси *z* середины отрезка между плоскостью отсечения и плоскостью проецирования, а затем поделить на длину этого отрезка и умножить на –2, чтобы получить значение 1 на плоскости отсечения, и –1 на плоскости проецирования:

$$z_{p,n} = -2 \cdot \frac{z_p - \frac{(-farVal) + (-nearVal)}{2}}{nearVal - farVal}$$

Точки z_p , лежащие либо за плоскостью отсечения, либо перед плоскостью проецирования, нормализованные по этой формуле, будут иметь координату $z_{p,n} \notin [-1; 1]$ и, следовательно, отображаться на экране компьютера не будут. Заметим, что *farVal* и *nearVal* могут иметь любые знаки.

Вышеупомянутые преобразования параллельного проецирования с нормализацией могут быть представлены в виде умножения вектора-точки справа на матрицу проецирования следующего вида:

$$\begin{bmatrix} \frac{2}{right - left} & 0 & 0 & -\frac{right + left}{right - left} \\ 0 & \frac{2}{top - bottom} & 0 & -\frac{top + bottom}{top - bottom} \\ 0 & 0 & -\frac{2}{farVal - nearVal} & -\frac{farVal + nearVal}{farVal - nearVal} \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Генерация такой матрицы с умножением ее на текущую матрицу осуществляется вызовом функции

void glOrtho(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble nearVal, GLdouble farVal)

с соответствующими параметрами.

4.3. Перспективное проецирование

При перспективном проецировании считается, что наблюдатель находится в точке (0, 0, 0) глобальной системы координат и смотрит в сторону убывания оси z . В таком случае значение координат по оси x относительно него возрастает вправо, а по оси y – вверх. Аналогично определяются плоскости отсечения и проекции, при этом $farVal > nearVal > 0$. На рис. 1 показан процесс перспективного проецирования.

Рассмотрим нахождение проекции (x_p, y_p, z_p) некоторой точки (x_0, y_0, z_0). Нахождение координат x_p, y_p принципиально не отличается, поэтому подробно рассмотрим эту процедуру только для y_p (рис. 2).

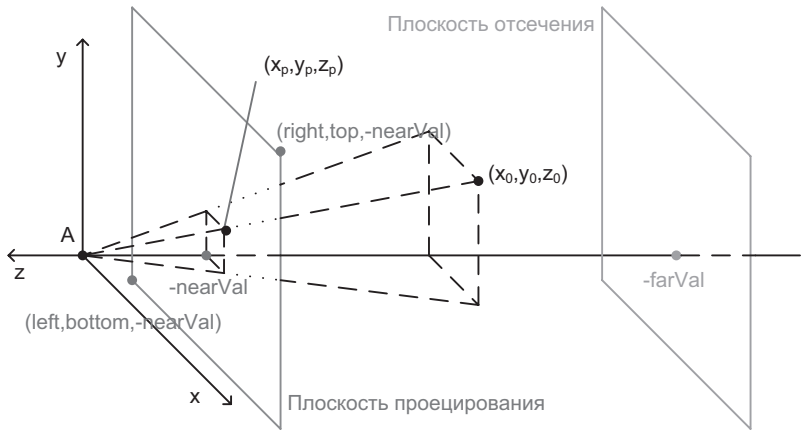


Рис. 1. Нахождение проекции точки (точка A – глаз наблюдателя)

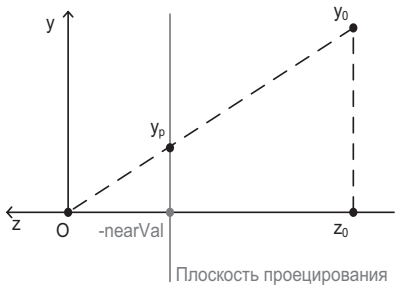


Рис. 2. Нахождение координаты проекции y_p точки с координатой y_0

Треугольники $0, y_p, -nearVal$ и $0, y_0, z_0$ подобны, следовательно, $\frac{y_p}{-nearVal} = \frac{y_p}{z_0}$, откуда $y_p = y_0 \cdot \frac{1}{z_0} \cdot (-nearVal)$, где умножение на $\frac{1}{z_0}$ носит название перспективного деления. Аналогично и для x: $x_p = x_0 \cdot \frac{1}{z_0} \cdot (-nearVal)$.

Кроме того, x_p, y_p должны быть нормализованы, и это должно быть учтено при составлении матрицы проецирования.

Матричное умножение невозможно совместить с делением на одну из координат вектора, поэтому в OpenGL используется 4-я координата точки. А именно первые три координаты вектора-произведения делятся на 4-ю, в качестве которой берется число $(-z_0)$.

Заметим также, что операция перспективного деления на 4-ю координату выполняется и в случае ортогональной проекции, но z_0 не влияет на это, так как матрица проецирования построена нужным образом.

Как было сказано выше, перспективному делению подвергается и координата глубины z_p, но при этом она также должна быть нормализована и лежать в интервале [–1, 1]. Для этого в OpenGL применяется следующая матрица перспективного проецирования, учитывающая все вышеупомянутые особенности:

$$\begin{bmatrix} \frac{2 * nearVal}{right - left} & 0 & \frac{right + left}{right - left} & 0 \\ 0 & \frac{2 * nearVal}{top - bottom} & \frac{top + bottom}{top - bottom} & 0 \\ 0 & 0 & \frac{farVal + nearVal}{farVal - nearVal} & \frac{2 * farVal * nearVal}{farVal - nearVal} \\ 0 & 0 & -1 & 0 \end{bmatrix}.$$

Генерация такой матрицы с умножением ее на текущую матрицу осуществляется вызовом функции

void glFrustum (GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble nearVal, GLdouble farVal)

с соответствующими параметрами.

4.4. Особенности проецирования

Следует отметить две важные особенности выбора параметров матриц проецирования.

1. При вычислении глубины точек-проекций может получиться так, что две точки окажутся ближе друг к другу, чем некоторое ϵ и у них совпадут все три координаты $(x_{p,n}, y_{p,n}, z_{p,n})$. Это явление называется конфликтом глубины (англ. *z-fighting*) и может повлечь нежелательные дефекты визуализации. Оно связано с ограниченной длиной разрядной сетки, выделяемой для представления чисел в памяти компьютера. Чтобы уменьшить ϵ, необходимо, чтобы значение величины $\frac{farVal}{nearVal}$ было как можно меньше, но при этом теряется приблизительно $\log_2 \frac{farVal}{nearVal}$ битов информации о глубине точки.

2. При выборе *nearVal* и вершин прямоугольника наблюдения требуется следить, чтобы соотношение его сторон и взаимное расположение с глазом наблюдателя как можно лучше соответствовало взаимному расположению глаз реального пользователя и его монитора с соответствующим соотношением сторон, иначе изображение будет неудобно наблюдать или же оно будет выглядеть вовсе неестественно.

4.5. Концепция наблюдателя при проецировании

Рассмотренных выше методов проецирования может быть недостаточно, когда необходим обзор сцены с произвольной точки зрения. Для этого вводится несколько дополнительных понятий.

Наблюдатель – совокупность глаза наблюдателя и связанной с ним системы отсчета. То есть наблюдателя можно описать матрицей наблюдения модели. Рассмотрим особенности проецирования объектов относительно произвольного наблюдателя:

1. перемещение наблюдается на некоторый вектор $\vec{d}$ эквивалентно перемещению всех объектов наблюдения на вектор $-\vec{d}$ (в глобальной системе отсчета);
2. применение преобразования поворота к наблюдателю эквивалентно применению поворота в противоположную сторону к глобальной системе координат.

Как рассматривалось выше, после преобразования матрицы наблюдения модели точки объектов наблюдения заданы в глобальной системе координат, а значит, точки всех объектов наблюдения с координатами в глобальной системе в совокупности можно рассматривать как один глобальный объект наблюдения – *сцену*.

Преобразование наблюдателя – преобразование заданных в глобальной системе отсчета точек всех объектов наблюдения (сцены) таким образом, чтобы проецирование производилось относительно наблюдателя.

Матрица преобразования наблюдателя – матрица, реализующая преобразование наблюдателя.

В OpenGL преобразование наблюдателя реализуется умножением заданной матрицы проецирования на матрицу преобразования наблюдателя: таким образом это преобразование повлияет на все проецируемые точки. Это действие может быть реализовано вызовом функции библиотеки GLU:

void gluLookAt(GLdouble eyeX, GLdouble eyeY, GLdouble eyeZ, GLdouble centerX, GLdouble centerY, GLdouble centerZ, GLdouble upX, GLdouble upY, GLdouble upZ).

Координаты (*eyeX; eyeY; eyeZ*) задают положение наблюдателя, (*centerX; centerY; centerZ*) – координаты точки, находящейся перед наблюдателем, (*upX; upY; upZ*) – вектор, направленный относительно наблюдателя вверх. Имея матрицу наблюдения модели, описывающую наблюдателя

$$\begin{pmatrix} x_{в_1}, & x_{в_2}, & x_{в_3}, & x_t \\ y_{в_1}, & y_{в_2}, & y_{в_3}, & y_t \\ z_{в_1}, & z_{в_2}, & z_{в_3}, & z_t \\ 0, & 0, & 0, & 1 \end{pmatrix}.$$

параметры функции можно определить следующим образом:

$$\begin{pmatrix} eyeX \\ eyeY \\ eyeZ \end{pmatrix} = \begin{pmatrix} x_t \\ y_t \\ z_t \end{pmatrix}, \qquad \begin{pmatrix} centerX \\ centerY \\ centerZ \end{pmatrix} = \begin{pmatrix} x_t + x_{в_3} \\ y_t + y_{в_3} \\ z_t + z_{в_3} \end{pmatrix}, \qquad \begin{pmatrix} upX \\ upY \\ upZ \end{pmatrix} = \begin{pmatrix} x_{в_2} \\ y_{в_2} \\ z_{в_2} \end{pmatrix},$$

где предполагается, что направление вверх совпадает с направлением оси y (вектора $\vec{e_2}$), а направление вперед – с осью *z* (вектор $\vec{e_3}$). В действительности не важно, как именно выбирается ориентация вверх, влево – этот выбор всегда относителен, главное – правильно придерживаться выбранной ориентации при разработке программного обеспечения.

Другие возможности моделирования

Выше были рассмотрены с теоретической и прикладной точек зрения положения, которые могут считаться базовыми для визуализации трехмерной графики с использованием инструментария

OpenGL. Ниже рассматриваются еще некоторые полезные возможности.

5.1. Определение параметров примитивов

Некоторые примитивы имеют дополнительные параметры:

- размер точки (GL_POINTS), режим интерполяции (GL_POINT) и др. параметры в единицах измерения буфера кадров (т. е. вне зависимости от удаленности точки от наблюдателя при перспективном проецировании) указываются при помощи вызова функции
  void glPointSize(GLfloat size);
- толщина линии (GL_LINES, GL_LINE_STRIP/LOOP), режим интерполяции GL_LINE) и др. параметры в единицах буфера кадров указываются при помощи вызова функции
  void glLineWidth(GLfloat width);
- фактура линии задается функцией
  void glLineStipple(GLint factor, GLushort pattern),

где pattern – 16-битовый шаблон, 1 в котором соответствует наличию точки при интерполяции, а 0 – отсутствию; шаблон применяется циклически, младшие биты учитываются сначала. Параметр factor определяет, сколько раз будет учтен каждый из битов шаблона, перед тем как перейти к следующему биту. Эту возможность нужно предварительно разрешить вызовом glEnable(GL_LINE_STIPPLE).

5.2. Использование цвета

В компьютерной графике принято кодировать цвет в виде разложения по яркостям трех базовых цветов: красного, зеленого, синего. В OpenGL цвета кодируются числами с плавающей точкой в диапазоне [0; 1]. Чем больше число, кодирующее компоненту, тем весомее ее вклад в формирование результирующего цвета. Каждая точка имеет свой цвет, а также дополнительную компоненту – так называемый альфа-канал (alpha), характеризующий прозрачность этой точки.

Чтобы использовать альфа-канал, нужно задействовать смешивание вызовом функции glEnable(GL_BLEND). В этом случае если новая точка после проецирования (source) имеет глубину более приоритетную, чем та, что уже находится в буфере кадров (destination), то можно вычислить результирующий цвет в зависимости от цветов обеих точек. Для этого нужно определить функцию смешивания. Вызов функции glBlendFunc(GL_SRC_ALPHA,

GL_ONE_MINUS_SRC_ALPHA) определит следующий порядок определения конечного цвета:

$$\begin{pmatrix} red \\ green \\ blue \end{pmatrix}_{final} = \begin{pmatrix} red \\ green \\ blue \end{pmatrix}_{source} \cdot A_{source} + \begin{pmatrix} red \\ green \\ blue \end{pmatrix}_{destination} \cdot (1 - A_{source})$$

где A_{source} – величина альфа-канала новой точки.

Смешивание позволяет добиться таких эффектов, как прозрачность материалов и отражения. Важно, что смешивание происходит только в том случае, если точка признана более приоритетной по глубине, иначе она не будет учтена.

Теперь рассмотрим задание цветов. Чтобы задать цвет фона – начального заполнения буфера изображения, используется функция

void glClearColor(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha),

чтобы очистить фон указанным цветом, вызывается функция

void glClear(GL_COLOR_BUFFER_BIT).

Также можно определить цвет для последующих добавленных вершин многоугольников при помощи вызова функции

void glColor4f(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha).

Эта функция имеет много вариантов. Интерполяция цвета при выполнении интерполяции точек многоугольника осуществляется вызовом функции

void glShadeModel(GLenum mode).

Если mode = GL_FLAT, то все точки будут интерполированы одним цветом – цветом последней вершины многоугольника (кроме GL_POLYGON, для него – первой), если mode = GL_SMOOTH, то цвет каждой точки будет интерполироваться с учетом цветов всех вершин.

Вывод трехмерной графики

Однако недостаточно получить НЭК для того, чтобы увидеть изображение на экране компьютерного монитора. Вывод графики в OpenGL осуществляется в *буфер кадров* (англ. frame buffer). Буфер вывода кадров представляет собой развернутый массив выровненных по сетке точек с заданным цветом, по аналогии с массивом пикселей на мониторе. И хотя буфер кадров является чаще всего линейным (одномерным) участком оперативной памяти компью-

тера, его целесообразно рассматривать как двухмерную таблицу с содержащими цвета точек ячейками, развернутую в зависимости от архитектуры компьютера, а точнее, от способа организации его оперативной памяти. В буфере кадров также задана такая система координат, что точка (0, 0) – это нижний левый угол, ось x направлена вправо, а ось y – вверх.

Исторически сложилось так, что система координат на мониторе компьютера, а также при программировании оконных интерфейсов (в частности, это касается координат указателя мыши) такова, что ось y возрастает вниз, а точка (0, 0) соответствует верхнему левому углу монитора (или окна). Перейти от адреса ячейки в буфере кадров к координатам окна можно, используя формулу

$$y_{окно} = (\text{Высота окна}) - y_{буфер}.$$

6.1. Понятие экрана

Чтобы поместить точку в буфер кадров, нужно преобразовать ее НЭК в координаты буфера. Чтобы описать этот процесс, определим *экран* как прямоугольную подобласть буфера кадров, соответствующую прямоугольнику проецирования. Экран определяется положением своего нижнего левого угла (x, y) в буфере кадров, а также своими шириной width и высотой height.

В итоге координаты точки $(x_{p,w}, y_{p,w})$ в буфере кадров вычисляются из НЭК $(x_{p,n}, y_{p,n})$ по формуле

$$(x_{p,w}, y_{p,w}) = \left((x_{p,n} + 1) \cdot \frac{width}{2}, \quad (y_{p,n} + 1) \cdot \frac{heigth}{2} \right).$$

Для задания экрана используется функция

void glViewport(GLint x, GLint y, GLsizei width, GLsize height)

с соответствующими параметрами.

6.2. Оконные приложения

Чтобы использовать интерфейс OpenGL в конкретной операционной системе (ОС), требуется учитывать особенности ее API. Настоящая статья не ставит целью описание всех таких особенностей. Один из самых удобных и часто используемых приемов реализации пользовательского интерфейса при работе с OpenGL – это использование кроссплатформенной библиотеки GLUT[5].

GLUT позволяет выводить графику в окно пользовательского интерфейса ОС. При этом клиентская часть окна, а именно графический буфер окна, ставится в соответствие (или физически при-

равнивается) буферу кадров OpenGL. То есть изображение, выведенное OpenGL в буфер кадров, будет отображено в окне.

Важной особенностью GLUT является поддержка двойной буферизации, при которой буфер кадров OpenGL и графический буфер окна – это два разных буфера. Тем самым OpenGL позволяет рисовать сложную (по затратам времени) сцену в буфер кадров и только в конце переключить буферы (поменяться с текущим окном). В итоге пользователь увидит на мониторе законченное изображение, а не процесс работы, который при динамической перерисовке сцены является нежелательным эффектом. В частности, перерисовка непосредственно в графическом буфере окна, как правило, приводит к заметному мерцанию экрана.

6.3. Применение буфера выбора

OpenGL может выводить не только изображение и не только в буфер кадров. Можно назначить имена (names) рисуемым примитивам, чтобы в дальнейшем получить список имен тех примитивов, которые при проецировании попадают в прямоугольник проецирования.

Вывод списка имен примитивов осуществляется в *буфер выбора*. Этот буфер необходимо указать в первую очередь вызовом функции

void glSelectBuffer(GLsizei size, GLuint *buffer),

где buffer – указатель на предварительно выделенный под буфер массив, size – максимально допустимое количество значений в массиве. Количество выделенной памяти рассчитывается в соответствии с форматом буфера (табл. 1) и количеством имен.

Указав буфер, можно перейти в режим выбора при помощи вызова функции

GLint glRenderMode(GLenum mode),

где mode = GL_SELECT соответствует режиму выбора, а mode = GL_RENDER (рендеринг) – режиму вывода графики в буфер кадров. После каждого перехода в режим выбора заполнение буфера выбора начинается сначала.

Перейдя в режим выбора, нужно инициализировать пустой стек имен вызовом функции glInitNames(). Далее можно помещать в стек значения :

void glPushName(GLuint name);

заменять значение, находящееся на вершине:

void glLoadName(GLuint name);

или удалять значения из вершины стека:

void glPopName(void).

После подготовки содержимого стека к отображению конкретного примитива или нескольких примитивов вывод изображения выполняется таким же образом, как и в режиме вывода графики. Если на момент следующего изменения стека имен хотя бы один рисуемый примитив попал в пределы прямоугольника наблюдения, то текущее наполнение стека будет скопировано в буфер выбора.

Модель именования в виде стека позволяет организовать иерархический выбор. Например, низ стека может соответствовать всей модели, а вершина стека – какой-либо ее отдельной части. Таким образом можно разделить перекрывающиеся имена.

Для завершения отображения всех необходимых объектов вызывается функция glRenderMode с параметром, отличным от GL_SELECT. Тогда эта функция вернет количество записей (но не значений!), попавших в буфер, но если размер буфера выбора оказался недостаточным, функция вернет отрицательное значение. Если хотя бы один примитив, отображенный с момента последнего изменения стека, попал в прямоугольник проецирования, то последнее наполнение стека также будет скопировано в буфер.

Наконец, содержимое буфера выбора обрабатывается в соответствии с форматом, приведенным в табл. 1.

Если требуется выбрать только объекты, которые бы попали в некоторый участок буфера кадров, то следует соответствующим образом изменить прямоугольник наблюдения, переопределив матрицу проецирования.

Таблица 1

Формат буфера выбора

Смещение*	**Содержимое буфера (значения)**
0	Количество имен в стеке имен n_1 первой записи
1	Минимальное значение нормированной глубины среди вершин объекта, умноженное на $2^{32} - 1$
2	Максимальное значение нормированной глубины
3	Значение, взятое с низа стека имен
...	Значения стека в порядке убывания глубины
$2 + n_1$	Значение, взятое с вершины стека
$3 + n_1$	Количество имен в стеке имен n_2 второй записи
...	...

*Смещение отсчитывается в величинах, принятых для типа GLuint.

6.4. Обратная связь

Помимо вывода графики и имен выбранных объектов, возможен вывод в буфер обратной связи информации о параметрах объектов сцены после обработки в буфер обратной связи (режим GL_FEEDBACK).

Для обнаружения объекта в буфере обратной связи он предварительно помечается так называемым токеном (англ. token) при помощи вызова функции

void glPassThrough(GLfloat token).

Это должно быть сделано до отрисовки примитивов соответствующего объекта.

Буфер обратной связи назначается функцией

void glFeedbackBuffer(GLsizei size, GLenum type, GLfloat *buffer), где параметр type определяет, какая информация должна попасть в буфер, при этом параметры size и buffer определяются аналогично тому, как это делалось для буфера выбора.

Для примера рассмотрим получение двухмерных координат в системе координат буфера кадров, которые бы имели примитивы после обработки в режиме вывода графики. В этом случае type = GL_2D. Размер буфера следует рассчитывать в соответствии с форматом буфера, его типом и количеством примитивов.

Рассмотрим формат буфера. Буфер состоит из записей, непрерывно следующих друг за другом. Каждая запись начинается с идентифицирующего ее токена, который является целочисленной константой, приведенной к типу GLfloat. Формат записей приведен в табл. 2.

Под вершиной понимается структура данных, соответствующая параметру type буфера. При type = GL_2D эта структура может быть определена следующим образом:

typedef struct {GLfloat x; GLfloat y;}VECTOR2F.

Назначив буфер обратной связи, переходят к предусмотренному в OpenGL режиму обратной связи, применяя функцию glRenderMode. Далее помеченные токенами при помощи функции glPassThrough объекты отображаются так, как если бы они записывались в буфер кадров, после чего следует переключиться в другой режим и обработать содержимое буфера.

Функция glRenderMode при выходе из режима обратной связи вернет количество попавших в буфер значений (но не записей!) или отрицательную величину, если размера буфера оказалось недостаточно.

Таблица 2

Формат записей буфера обратной связи

Токен	**Содержимое**			
`GL_POINT_TOKEN` (примитив точка)	вершина	(следующая запись)		
`GL_LINE_TOKEN` `GL_LINE_RESET_TOKEN` (примитив отрезок)	вершина	вершина	(следующая запись)	
`GL_POLYGON_TOKEN` (примитив многоугольник)	количество вершин *n*	вершина 1	...	вершина *n*
`GL_PASS_THROUGH_TOKEN` (метка, назначенная вызовом `glPassThrough`)	значение метки `token`	(следующая запись)		
`GL_DRAW_PIXEL_TOKEN` `GL_COPY_PIXEL_TOKEN` `GL_BITMAP_TOKEN` (прочие типы записей*)	вершина	(следующая запись)		

* Рассмотрение прочих типов записей выходит за рамки настоящей работы, но их все же необходимо корректно обрабатывать при разборе буфера обратной связи.

Заключение

В статье обобщены задачи визуализации данных и процессов, возникающие при разработке программного обеспечения с использованием кроссплатформенного интерфейса прикладного программирования OpenGL, а также некоторые практические приемы их программной реализации. К таким задачам относятся прежде всего геометрическое описание трехмерных объектов, выполнение матричных операций, преобразования проецирования объектов, управление цветом и интерполяцией геометрических фигур, вывод трехмерной графики на экран и в буферы видеокарты, а также ряд других вспомогательных задач. Проведенный анализ позволяет сделать вывод о том, что интерфейс прикладного программирования OpenGL предоставляет весьма широкий и полный набор функций для визуализации трехмерных объектов, статических сцен и интерактивных процессов.

Несмотря на определенные сложности и специфику приемов управления визуализацией, реализуемых библиотеками на основе OpenGL, применение интерфейса OpenGL предоставляет несомненные преимущества разработчикам сложных программных комплексов, требующих интенсивного использования функций компьютерной графики. К основным преимуществам следует от-

нести кроссплатформенность, позволяющую обеспечить переносимость прикладных программ между многими аппаратно-программными платформами, возможность разработки программ на разных языках программирования в пределах одного проекта, унификацию способов передачи данных между модулями программ, поддержку вывода изображения высокой четкости в реальном масштабе времени без потери качества изображения. Интерфейс OpenGL практически не имеет конкурентов, за исключением интерфейса DirectX на платформе Windows.

Став одним из ведущих инструментов реализации компьютерной графики и анимации, спецификация OpenGL в настоящее время является стандартом де-факто и обладает большим потенциалом дальнейшего развития. В связи с этим реализации OpenGL могут быть рекомендованы в качестве основного рабочего инструментария широкому кругу разработчиков систем компьютерной графики и научной визуализации.

Примечания

[1] См.: *Шикин Е.В., Плис А.И.* Кривые и поверхности на экране компьютера. Руководство по сплайнам для пользователей. М.: Диалог-МИФИ, 1996. 240 с.

[2] См.: *Shreiner D., Sellers G., Kessenich J., Licca-Kane B.* OpenGL Programming Guide. 8th ed.: The Official Guide to Learning OpenGL, Version 4.3. Upper Saddle River, NJ: Addison-Wesley, 2013. 986 p.; *Wright R., Haemel N., Sellers G., Lipchak B.* OpenGL Superbible. 5th ed.: Comprehensive Tutorial and Reference. Boston, MA: Addison-Wesley, 2011. 1002 p.

[3] См.: OpenGL 4 References Pages [Электронный ресурс] // Сайт проекта OpenGL. URL: http://www.opengl.org/sdk/docs/man (дата обращения: 09.03.2014).

[4] См.: The OpenGL Utility Toolkit [Электронный ресурс] // Сайт проекта OpenGL. URL: http://www.opengl.org/resources/libraries/glut (дата обращения: 09.03.2014).

[5] См.: The OpenGL Utility Toolkit (GLUT) Programming Interface API Version 3 [Электронный ресурс] // Сайт проекта OpenGL. URL: http://www.opengl.org/resources/libraries/glut/glut-3.spec.pdf (дата обращения: 09.03.2014).

Abstracts

V. Grigoriev, L. Schurkin

NET-CENTRIC WARFARE FROM SYNERGETICS POSITION

Today one of the most urgent problems in the field of ensuring the security of the state is to study issues of sustainability to the possible impacts of a potential enemy. The proposed new effective methods of modelling of information confrontation and conflicts of various scale and intensity, with the aim of identifying ways and directions of deterrence so-called "network-centric warfare" (NCW), reduce risk when making decisions in crisis situations.

Key words: network society, the information confrontation, complex system, stability, synergetics, complex network.

D. Ivanov, A. Nikitin

COUNTERACTION AGAINST KEYBOARD HANDWRITING ANALYSIS

In this work the authors propose method to protect the user from identifying him by his keyboard handwriting. Both software and hardware implementations of the security module, proposed to protect from the analysis of handwriting keyboard, are reviewed. There are proposed two concepts of the implementing module, which is placed between the keyboard and the computer. Schematic diagram for constructing a hardware module for PS/2 port is also shown.

Key words: keyboard handwriting, user identification, anonymization.

D. Larin

CRYPTOGRAPHIC OPERATIONS IN THE CRIMEAN WAR PERIOD

160 years ago, on March 28, 1854 Britain and France declared war against the Russian Empire, thus seriously altering the forces balance in the next Russian-Turkish confrontation, which began in 1853. The causes of the war was a reluctance on the part of European powers to strengthen Russia, which is practically defeated Ottoman Empire. The foregoing powers entered the war on the side Turkey, and later joined the Sardinia Italy Kingdom. Russian forces launched a coalition of a number of defeats in the Caucasus, Kamchatka, in the St. Petersburg district, as well as the Solovetsky Islands, but, unfortunately, in the main theatre of war in the Crimea, we were defeated.

From the point of view of information security, this war was the first in which the principally new telecommunication means (the Telegraph) were applied to control the fighting. Our military is the first who used the Telegraph to send encrypted messages in the war period.

Key words: Crimean war, cryptography, encryption, decryption, cipher, telegraph, communication.

A. Lavrentiev

OPTIMIZATION OF MULTI-COMPONENT BASED APPLICATIONS IN CLOUD ENVIRONMENT WITH MULTIPLE CLOUD PROVIDERS

In this work two-steps hybrid optimization approach is considered for implementation of multi-component application in cloud environment that consists of multiple clouds. MILP-based method is proposed in order to automatize selection of cloud providers, split workload between them, and optimize cloud architectures with goal of cost minimization of cloud resources usage while guarantying adequate QoS.

Key words: multi-cloud, multi-component application, queuing theory, linear programming, local search.

A. Malyuk, N. Miloslavskaya

TOWARDS THE INFORMATION SECURITY THEORY

On the experience of the Russian Federation, the development history of research aimed at the Information Security (IS) Theory

scientific and methodological foundations creation is examined in the article. The results obtained by the authors in the development of the informal systems theory and approaches to the IS processes simulation models creation in conditions of input data incomplete and insufficient reliability are presented. Based on the developed structure of a unified IS concept the issues arising during the practical implementation of an integrated protection system are considered consequentially.

Key words: information security theory, information security concept, knowledge autoformalization, information security models, information protection assessment, requirements to information protection, information protection systems.

N. Martynov, O. Kazarin

SUBTHRESHOLD CHANNELS AND PROTECTION METHODS FROM THEIR CREATION IN INTERACTIVE IDENTIFICATION SCHEMES

The article is devoted to the analysis of the subthreshold channels properties. The possibility of broadband subthreshold channels existence in digital signature schemes of domestic standards GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.10-2012 and interactive user identification schemes is considered. Special attention is paid to the formation principles of broadband subthreshold channels and protection methods from them.

Subthreshold channels can be considered as the steganographic channels created in variety of cryptographic constructions, where some random parameters are replaced with subthreshold messages, specially generated by potential intruders.

Key words: subthreshold channel, broadband subthreshold channel, digital signature scheme interactive identification.

S. Petrov, A. Tarasov

CULTURE DIGITAL HERITAGE: PROBLEMS OF FORMATION, DEVELOPMENT AND SECURITY

Some approaches of the Humanities and Computer science to the problems of formation, preservation, access and information security culture digital heritage are considered. The process approach to the

description of the digital heritage as a cultural object model and the digital heritage as a complex socio-technical system is proposed. Information systems by type of various cultural values and types cultural activities, as well as the content of information security are described and analysed.

Key words: digital heritage, cultural heritage, socio-technical system, information system, information security.

A. Platonova

INFORMATION SYSTEM OF THE EDUCATIONAL ACTIVITIES MULTIVARIABLE CONTROL

One of stages of creation of any information system is development of architecture, the user interface and programs for possibility of practical use of all modules of system. The choice of technical solutions at creation of information system of multiple parameter control of educational activity is made taking into account specifics of concrete subject domain – control and estimated activity in education and need of carrying out multiple parameter monitoring at the level of the territorial subject of the Russian Federation. The block diagram and the user interface are as a result created and algorithms of control and formation of a multiple parameter assessment of the pupil are programmatically realised.

Key words: information system, control, educational activity, architecture, user interface, program realisation.

A. Pupykina, A. Satunina

WEB APPLICATION METAMODEL FORMALIZATION

The article discusses a way of formalizing the web application metamodel for usage in model-driven development. The specifics of the proposed approach to designing web applications is the development and use of formal modeling method based on the development of graph representations.

Key words: model, model driven approach, transformation, web application.

A. Satunina, L. Sysoeva

METHODOLOGICAL ASPECTS OF METRICS SYSTEM FORMATION DURING INFORMATION SYSTEM PROJECT IMPLEMENTATION

The article describes the methodology used in the process of project realization on service-oriented information systems (IS) creation and influencing the metrics formation. Currently implementing IS projects are characterized by the fact that already at the initial stages of the project it is necessary to provide the ways and methods of measurement appropriate indicators and metrics that will be needed for the implemented system integration with BI-system. In the result of the analysis of the multilevel structure metrics information systems, service-oriented architecture was formed, designed with the RUP methodologies, ITIL, ITSM, BPM, project management.

Key words: process approach; project management of IP; service-oriented architecture, governance, service-oriented information system; metric information system.

N. Tarasova

ORGANIZATIONAL RISKS STRUCTURIZATION IN THE COMPANIES INFORMATION SECURITY SYSTEMS

The article deals with the content and causes of organizational risks in company information security systems. It provides the classification of organizational risks. Organizational risks in company information security systems are structured as in FTA and FA.

Key words: information security system, organizational risks, risk structure.

S. Zapechnikov

DATA AND PROCESSES VISUALIZATION BY OPENGL CROSS-PLATFORM PROGRAMMING INTERFACE

The paper is about data and processes visualization represented as 3D-objects, scenes or animated images by cross-platform programming libraries based on OpenGL interface. The mathematical and technical tasks, which can be solved using the OpenGL interface and libraries,

are determined, for instance, tasks of 3D-objects geometric description, their projecting on the screen, video memory management during the 3D-objects drawing etc. The tasks are organized into a system, for most of them the formal definitions are given, and the solving methods are pointed out. Practical recommendations for rational using of OpenGL interface in 3D-graphics programming are offered.

Key words: data visualization, 3D-graphics, geometric mappings, application programming interface, cross-platform software, OpenGL specification.

S. Zheltov

GENERAL METHODS OF COMPUTER SYSTEMS PRACTICAL EVALUATION AGAINST RISKS OF INFORMATION SECURITY THREATS IMPLEMENTATION IN THEIR USAGE

The article is devoted to some aspects of the technical evaluation (computing) capabilities potential offender information security of automated systems.

Key words: information security threats, security risks, automated systems.

Сведения об авторах

Григорьев Виталий Робертович – кандидат технических наук, главный научный консультант ЗАО «РНТ», grigorjev_vr@mail.ru

Желтов Сергей Александрович – старший преподаватель кафедры компьютерной безопасности и математических методов управления Тверского государственного университета, zheltov_s@mail.ru

Запечников Сергей Владимирович – доктор технических наук, доцент, профессор кафедры информационной безопасности банковских систем Национального исследовательского ядерного университета «МИФИ», svzapechnikov@mephi.ru

Иванов Дмитрий Александрович – студент Московского государственного технического университета радиотехники, электроники и автоматики (МГТУ МИРЭА), fenix.104.2@mail.ru

Казарин Олег Викторович – доктор технических наук, ведущий научный сотрудник отдела математических проблем информационной безопасности Института проблем информационной безопасности Московского государственного университета им. М.В. Ломоносова, okaz2005@yandex.ru

Лаврентьев Александр Владимирович – аспирант Московского государственного технического университета радиотехники, электроники и автоматики (МГТУ МИРЭА), lavrentev@hush.com

Ларин Дмитрий Александрович – кандидат технических наук, доцент кафедры интеллектуальных технологий и систем Московского государственного технического университета

радиотехники, электроники и автоматики (МГТУ МИРЭА), greattzar@yandex.ru

Малюк Анатолий Александрович – кандидат технических наук, профессор, профессор кафедры компьютерной безопасности Национального исследовательского ядерного университета «МИФИ» и кафедры информационной безопасности Финансового университета при Правительстве РФ, aamalyuk@mephi.ru

Мартынов Никита Романович – аспирант Института информационных наук и технологий безопасности РГГУ, nmartynov88@gmail.com

Милославская Наталья Георгиевна – кандидат технических наук, доцент кафедры информационной безопасности банковских систем Национального исследовательского ядерного университета «МИФИ»

Никитин Андрей Павлович – аспирант Московского государственного технического университета радиотехники, электроники и автоматики (МГТУ МИРЭА), gouststalker@mail.ru

Петров Сергей Томасович – главный редактор журнала «Цифровое наследие», 5008604@gmail.com

Платонова Алла Сергеевна – соискатель кафедры «Физика и прикладная математика» Муромского института (филиала) Владимирского государственного университета им. Александра Григорьевича и Николая Григорьевича Столетовых, allaplatonova@inbox.ru

Пупыкина Анна Александровна – аспирантка Института информационных наук и технологий безопасности РГГУ, anna.pupikina@gmail.com

Сатунина Анна Евгеньевна – кандидат экономических наук, ведущий научный сотрудник, декан факультета информатики Института информационных наук и технологий безопасности РГГУ, aesat@mail.ru

Сысоева Леда Аркадьевна – кандидат технических наук, доцент, директор Центра дистанционных технологий обучения РГГУ, leda@rggu.ru

Тарасов Александр Алексеевич – доктор технических наук, профессор, директор Института информационных наук и технологий безопасности РГГУ, aa_tarasov@list.ru

Тарасова Наталья Александровна – студентка кафедры комплексной защиты информации Института информационных наук и технологий безопасности РГГУ, taraso-natasha@yandex.ru

Шуркин Леонид Олегович – аспирант Московского государственного технического университета радиотехники, электроники и автоматики (МГТУ МИРЭА), bion2005@yandex.ru

General data about the authors

Grigoriev Vitaliy R. – Ph.D. in Engineering, main scientific consultant, CJSC "RNT", grigorjev_vr@mail.ru

Ivanov Dmitriy A. – student, Moscow State Technical University of Radio Engineering, Electronics and Automation, fenix.104.2@mail.ru

Kazarin Oleg V. – Dr. in Engineering, leading researcher, Department of Mathematical Problems of Information Security, Institute for Information Security Issues, Lomonosov Moscow State University, okaz2005@yandex.ru

Lavrentiev Alexander V. – postgraduate student, Moscow State Technical University of Radio Engineering, Electronics and Automation, lavrentev@hush.com

Larin Dmitriy A. – Ph.D. in Engineering, associate professor, Department of Intellectual Technologies and Systems, Moscow State Technical University of Radio Engineering, Electronics and Automation, greattzar@yandex.ru

Malyuk Anatoliy A. – Ph.D. in Engineering, professor, professor, Department of Computer Security, National Research Nuclear University "MEPhI"; professor, Department of Information Security, Financial University under the Government of the Russian Federation, aamalyuk@mephi.ru

Martynov Nikita R. – postgraduate student, Institute for Information Sciences and Security Technologies, Russian State University for the Humanities, nmartynov88@gmail.com

Miloslavskaya Natalia G. – Ph.D. in Engineering, associate professor, Department of Information Security of Banking Systems, National Research Nuclear University "MEPhI"

Nikitin Andrey P. – postgraduate student, Moscow State Technical University of Radio Engineering, Electronics and Automation, gouststalker@mail.ru

Petrov Sergey T. – editor-in-chief, magazine "Digital Heritage", 5008604@gmail.com

Platonova Alla S. – applicant, Department "Physics and Applied Mathematics", Murom Institute (branch), Vladimir State University named after Alexander and Nikolay Stoletovs, allaplatonova@inbox.ru

Pupykina Anna A. – postgraduate student, Institute for Information Sciences and Security Technologies, Russian State University for the Humanities, anna.pupikina@gmail.com

Satunina Anna E. – Ph.D. in Economics, leading researcher, dean, Faculty of Computer Science, Institute for Information Sciences and Security Technologies, Russian State University for the Humanities, aesat@mail.ru

Schurkin Leonid O. – postgraduate student, Moscow State Technical University of Radio Engineering, Electronics and Automation, bion2005@yandex.ru

Sysoeva Leda A. – Ph.D. in Engineering, associate professor, director, Centre for Distance Learning Technologies, Russian State University for the Humanities, leda@rggu.ru

Tarasov Alexander A. – Dr. in Engineering, professor, director, Institute for Information Sciences and Security Technologies, Russian State University for the Humanities, aa_tarasov@list.ru

Tarasova Natalia A. – student, Department of Complex Information Security, Institute for Information Sciences and Security Technologies, Russian State University for the Humanities, taraso-natasha@yandex.ru

Zapechnikov Sergey V. – Dr. in Engineering, associate professor, professor, Department of Information Security of Banking Systems, National Research Nuclear University "MEPhI", svzapechnikov@mephi.ru

Zheltov Sergey A. – senior lecturer, Department of Computer Security and Mathematical Methods of management, Tver State University, zheltov_s@mail.ru

Заведующая редакцией *И.В. Лебедева*
Художник *В.В. Сурков*
Художник *В.Н. Хотеев*
Корректор *О.К. Юрьев*
Компьютерная верстка *Я.Р. Качалова*

Формат 60×90 1/16.
Усл. печ. л. 14,2. Уч.-изд. л. 14,9.
Тираж 1050 экз. Заказ № 110

Издательский центр
Российского государственного
гуманитарного университета
125993, Москва, Миусская пл., 6
www.rggu.ru
www.knigirggu.ru