

# Программная инженерия

Пр 5  
2012  
ИН

Учредитель: Издательство "НОВЫЕ ТЕХНОЛОГИИ"

Издается с сентября 2010 г.

## Редакционный совет

Садовничий В.А., акад. РАН  
(председатель)  
Бетелин В.Б., акад. РАН  
Васильев В.Н., чл.-корр. РАН  
Жижченко А.Б., акад. РАН  
Макаров В.Л., акад. РАН  
Михайленко Б.Г., акад. РАН  
Панченко В.Я., акад. РАН  
Стемпковский А.Л., акад. РАН  
Ухлинов Л.М., д.т.н.  
Федоров И.Б., акад. РАН  
Четверушкин Б.Н., акад. РАН

## Главный редактор

Васенин В.А., д.ф.-м.н.

## Редколлегия:

Авдошин С.М., к.т.н.  
Антонов Б.И.  
Босов А.В., д.т.н.  
Гаврилов А.В., к.т.н.  
Гуриев М.А., д.т.н.  
Дзегеленок И.Ю., д.т.н.  
Жуков И.Ю., д.т.н.  
Корнеев В.В., д.т.н.,  
Костюхин К.А., к.ф.-м.н.  
Липаев В.В., д.т.н.  
Махортов С.Д., д.ф.-м.н.  
Назирова Р.Р., д.т.н.  
Нечаев В.В., к.т.н.  
Новиков Е.С., д.т.н.  
Норенков И.П., д.т.н.  
Нурминский Е.А., д.ф.-м.н.  
Павлов В.Л., д.ф.-м.н.  
Пальчунов Д.Е., д.т.н.  
Позин Б.А., д.т.н.  
Русаков С.Г., чл.-корр. РАН  
Рябов Г.Г., чл.-корр. РАН  
Сорокин А.В., к.т.н.  
Терехов А.Н., д.ф.-м.н.  
Трусов Б.Г., д.т.н.  
Филимонов Н.Б., д.т.н.  
Шундеев А.С., к.ф.-м.н.  
Язов Ю.К., д.т.н.

## Редакция

Лысенко А.В., Чугунова А.В.

Журнал издается при поддержке Отделения математических наук РАН, Отделения нанотехнологий и информационных технологий РАН, МГУ имени М.В. Ломоносова, МГТУ имени Н.Э.Баумана, ОАО "Концерн "Сириус".

## СОДЕРЖАНИЕ

<b>Галатенко В. А., Костюхин К. А., Шмырев Н. В., Аристов М. С.</b> Использование свободно распространяемых средств статического анализа исходных текстов программ в процессе разработки приложений для операционных систем реального времени . . . . .	2
<b>Корзун Д. Ж., Ломов А. А., Ваняг П. И.</b> Автоматизированная модельно-ориентированная разработка программных агентов для интеллектуальных пространств на платформе Smart-M3 . . . . .	6
<b>Костюк В. В., Бовкунович М. А.</b> Элементы сервис-ориентированной разработки архитектуры прикладного программного обеспечения на основе бизнес-моделей . . . . .	15
<b>Соловьев С. В., Затока И. В., Лавлинская О. П.</b> Особенности проектирования информационных систем обеспечения деятельности по технической защите информации . . . . .	21
<b>Жуков И. Ю., Михайлов Д. М., Кукушкин А. В., Стариковский А. В., Иванова Е. В., Федоров Е. Д., Толстая А. М.</b> Программные методы распознавания участков желудочно-кишечного тракта, полипов и кровотечений в беспроводной капсульной эндоскопии . . . . .	28
<b>Черемисина Е. Н., Белов М. А., Антипов О. Е., Сорокин А. В.</b> Инновационная практика компьютерного образования в университете "Дубна" с применением виртуальной компьютерной лаборатории на основе технологии облачных вычислений . . . . .	34
<b>Славин Б. Б.</b> Посткраудсорсинг как архитектура экспертных сетей . . . . .	42
<b>Contents</b> . . . . .	48

Журнал зарегистрирован

в Федеральной службе

по надзору в сфере связи,

информационных технологий

и массовых коммуникаций.

Свидетельство о регистрации

ПИ № ФС77-38590 от 24 декабря 2009 г.

Журнал распространяется по подписке, которую можно оформить в любом почтовом отделении (индексы: по каталогу агентства "Роспечать" — 22765, по Объединенному каталогу "Пресса России" — 39795) или непосредственно в редакции.

Тел.: (499) 269-53-97. Факс: (499) 269-55-10.

Http://novtex.ru E-mail: prin@novtex.ru

Журнал включен в систему Российского индекса научного цитирования.

Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

© Издательство "Новые технологии", "Программная инженерия", 2012

**В. А. Галатенко**, д-р физ.-мат. наук, зав. сектором, e-mail: galat@niisi.msk.ru,  
**К. А. Костюхин**, канд. физ.-мат. наук, стар. науч. сотр., e-mail: kost@niisi.msk.ru,  
**Н. В. Шмырев**, канд. физ.-мат. наук, науч. сотр., e-mail: shmyrev@niisi.msk.ru,  
**М. С. Аристов**, мл. науч. сотр., e-mail: maristov@niisi.ras.ru,  
НИИСИ РАН, г. Москва

# Использование свободно распространяемых средств статического анализа исходных текстов программ в процессе разработки приложений для операционных систем реального времени

*Современные операционные системы реального времени обладают определенной спецификой, связанной с особыми предъявляемыми к ним требованиями. В данной работе рассмотрены ошибки, характерные для операционных систем реального времени, а также методы диагностики и выявления этих ошибок, основанные на использовании средств статического анализа исходных текстов программ.*

**Ключевые слова:** статический анализ, отладка, операционные системы реального времени

## Введение

В НИИСИ РАН уже более 10 лет ведутся разработка и поддержка собственной линейки операционных систем реального времени — ОС РВ Багет 2.0 [1] и ОС РВ Багет 3.0 [2]. При этом используются как собственные средства разработки, так и свободно распространяемые. В частности, для статического анализа исходных текстов используются именно свободно распространяемые средства, применению которых и посвящена эта статья.

Следует заметить, что рассматриваемая далее специфика ОС РВ Багет характерна и для других ОС РВ, например ОС VxWorks [3].

## Специфика архитектуры ОС РВ Багет

В настоящем разделе рассмотрим отличительные особенности ОС РВ Багет 2.0 и 3.0 и связанные с ними потенциальные ошибки в программном коде.

*Общие ресурсы системных и пользовательских потоков управления.* С точки зрения стандарта POSIX ОС РВ Багет 2.0 представляет собой один процесс, поэтому потоки управления в ОС РВ Багет 2.0 работают в общей

памяти и не защищены друг от друга. Кроме того, общими являются и другие ресурсы, например файловые дескрипторы.

Типичными ошибками, возникновение которых может привести к необратимым последствиям для всей системы, в этом случае являются:

- выход за границу массива;
- использование больших локальных объектов (располагающихся в стеке потока), так как это может привести к переполнению стека;
- захват ресурсов без освобождения (если, например, явно не будет закрыт открытый ранее файл, то он останется открытым и после завершения потока управления, работавшего с этим файлом);
- использование неинициализированных переменных (особенно указателей);
- обращение по нулевому адресу.

*Особый режим выполнения обработчиков прерываний.* При написании кода критически важных системных объектов, таких как обработчики прерываний, необходимо соблюдать определенные правила кодирования. В частности, нельзя вызывать некоторые функции, необ-

ходимо запрещать прерывания и предпринимать другие подобные действия. Любое отклонение от этих правил может привести к возникновению ошибки.

*Следование стандарту ARINC 653.* ОС РВ Багет 3.0 в штатном режиме функционирования должна удовлетворять требованиям стандарта ARINC 653 [4] по надежности и отказоустойчивости системы. Это накладывает дополнительные ограничения на программный код пользовательских процессов ОС.

Так, например, использование функций создания потока управления или выделения динамической памяти следует трактовать как ошибку в программном коде.

### Применение средств статического анализа при разработке ОС РВ Багет и ее компонентов

В этом разделе будут рассмотрены два свободно распространяемых средства статического анализа кода, с помощью которых можно выявить практически все отмеченные выше ошибки.

Для выявления таких ошибок, как обращение по нулевому адресу, использование неинициализированных переменных и подобных им, синтаксического анализа исходного кода недостаточно. Инструмент статического анализа должен проводить семантический анализ, используя, например, граф потока управления программы (*Control Flow Graph*, CFG) и граф потока данных программы (*Data Flow Graph*, DFG). По этой причине авторы статьи остановили свой выбор на средствах статического анализа, интегрированных в компилятор.

В качестве последних использовались *Clang Static Analyzer* (CSA) [5], интегрированный в си-компилятор проекта LLVM (*Low Level Virtual Machine*)<sup>1</sup>, и *mygcc* [6], интегрированный в си-компилятор проекта GCC (*GNU Compiler Collection*)<sup>2</sup>.

*Clang Static Analyzer.* Статический анализатор Clang позволяет осуществлять поиск ошибок и недостатков программного кода различных типов. В их числе — наличие в коде неиспользуемых переменных, использование неинициализированных переменных, логические ошибки (ошибки вычисления счетчика цикла или индекса мас-

сива, ошибки обращения по нулевому указателю, неверные логические выражения), ошибки управления выделением динамической памяти. Кроме того, CSA поддерживает аннотирование кода в виде набора атрибутов (частично заимствованных из проекта GCC). При помощи таких аннотаций разработчик детализирует информацию о логике выполнения кода, позволяя средству статического анализа проводить более качественную проверку. Пример отчета, порожденного CSA, приведен на рис. 1.

Одним из достоинств CSA является простота его интеграции в процесс разработки. Если при сборке используется make-файл, то достаточно выполнить команду `scan-build make`, при необходимости указав нужный компилятор (по умолчанию используется gcc). В результате будет проведена не только сборка проекта, но и проверка его исходного кода.

Итоги проверки исходного кода ядра ОС РВ Багет 3.0 приведены на рис. 2.

Количество ложных диагностических предупреждений у CSA на данном примере составляет примерно

```
1 #ifndef KERNEL
2 #error Invalid defines
3 #endif
4 #include <try.h>
5 #include <string.h>
6 #include <private/kernel.h>
7 #include <arch/cpu.h>
8 #include <private/errset.h>
9 static void errorStackOverflow()
10 {
11     int a[1000];
12     a[1]=5;
13     errorStackOverflow();
14 }
15 int errorGen(error_type_t n)
16 {
17     int *a = 0;
18     int error=-1;
19     switch (n)
20     {
21     case errorTYPE_TLBMOD:
22         a[0] = 5;
23         break;
```

Рис. 1. Аннотированный html-отчет о выявленной ошибке

<sup>1</sup> <http://llvm.org>

<sup>2</sup> <http://gcc.gnu.org>

25 % от общего числа предупреждений. В основном, ложная диагностика появляется при попытке выявить потенциальные логические ошибки (раздел "Logic error" на рис. 2).

Из перечисленных выше ошибок CSA позволяет выявить следующие:

- выход за границу массива;
- использование неинициализированных переменных (особенно указателей);
- обращение по нулевому адресу.

*Анализатор mуссс.* Еще одним инструментом статического анализа, интегрированным в компилятор, является анализатор mуссс, часть экспериментальной разработки в рамках проекта GCC (ветвь Graphite), использующий для статического анализа внутреннее представление компилятора gcc. В отличие от CSA, mуссс требует предварительной настройки на исходный анализируемый код в виде внешних аннотаций, задаваемых как в командной строке, так и в отдельном файле. Для этих целей предусмотрены ключи компилятора `-free-check` и `-free-checks`. Пример аннотаций, при помощи которых mуссс анализирует корректность работы с прерываниями и наличие в коде запрещенной функции `pthread_create`, приведен на рис. 3.

Разберем листинг, представленный на рис. 3, более подробно. Дополнительные проверки в анализаторе mуссс осуществляются при помощи так называемых *condates* — правил, задающих начальную и конечную вершины на графе потока управления (CFG) и контрольные вершины между ними. Если анализатору mуссс удастся найти допустимый путь на CFG между начальной и конечной вершинами, не проходящий через контрольные вершины, то порождается соответствующее предупреждение.

Первое правило из листинга предназначено для проверки корректности работы с прерываниями, т. е. после запрета прерываний (вызов `intLock`) и до возврата из функции прерывания должны быть разрешены (вызов `intUnlock`). Аналогичным образом можно осуществлять проверку использования системных ресурсов (например, динамической памяти, файловых дескрипторов).

При помощи второго правила из листинга можно выявлять в исходном коде вызовы функций, запрещенных принятым стандартом кодирования.

В настоящее время идет доработка анализатора mуссс для создания правил с использованием языка, описывающего внутреннее представление программы в компиляторе. Для выявления больших локальных объектов раз-

## Bug Summary

Bug Type	Quantity	Display?
<b>All Bugs</b>	<b>269</b>	<input checked="" type="checkbox"/>
<b>Dead code</b>		
Idempotent operation	28	<input checked="" type="checkbox"/>
<b>Dead store</b>		
Dead assignment	135	<input checked="" type="checkbox"/>
Dead increment	11	<input checked="" type="checkbox"/>
Dead initialization	2	<input checked="" type="checkbox"/>
Dead nested assignment	20	<input checked="" type="checkbox"/>
<b>Logic error</b>		
Assigned value is garbage or undefined	12	<input checked="" type="checkbox"/>
Branch condition evaluates to a garbage value	1	<input checked="" type="checkbox"/>
Called function pointer is null (null dereference)	1	<input checked="" type="checkbox"/>
Dereference of null pointer	43	<input checked="" type="checkbox"/>
Division by zero	4	<input checked="" type="checkbox"/>
Function call argument is an uninitialized value	2	<input checked="" type="checkbox"/>
Garbage return value	1	<input checked="" type="checkbox"/>
Result of operation is garbage or undefined	7	<input checked="" type="checkbox"/>
<b>Unix API</b>		
Undefined allocation of 0 bytes	2	<input checked="" type="checkbox"/>

Рис. 2. Результат проверки исходного кода ядра ОС PV Baget 3.0 статическим анализатором Clang

```

condate missing_intUnlock {
    from "%X = intLock()" or "intLock()"
    to ("return" or "return %_")
    avoid "intUnlock (%Y)"
} warning("Unreleased interrupt lock");

condate pthread_create {
    "%X = pthread_create(%_)" or "pthread_create(%_)"
} warning("pthread_create() in arinc 653
applications");

```

Рис. 3. Пример аннотаций mуссс

работчики mygcc предлагают использовать конструкции типа [7]:

```
"%T %X"|TYPE_P(T) && TREE_CODE(X)==VAR_DECL
&& DECL_SIZE(X)>1024
```

Приведенное правило сработает, если размер локального объекта превысит 1 Кб.

При статическом анализе исходных текстов ядра ОС РВ Багет 3.0 с помощью анализатора mygcc и набора правил, содержащего проверки корректности использования ресурсов и работы с прерываниями, произошло шесть срабатываний анализатора. Из них два оказались ложными.

Следует отметить, что использование псевдоязыка для задания правил статического анализа дает анализатору mygcc большую гибкость при настройке на конкретный проект в сравнении с CSA, хотя и требует больше времени.

### Стандарт кодирования для ОС РВ

Как и во всех критически важных системах, цена ошибки в ОС РВ является очень высокой. В настоящее время существует достаточно много рекомендаций, оформленных в виде набора правил, стандартов кодирования, применяемых к ОС РВ, к встраиваемым системам и т. п., позволяющих повысить надежность разрабатываемых систем. Среди этих рекомендаций следует выделить рекомендации MISRA (Motor Industry Software Reliability Association [8]), на основе которых можно разработать собственный стандарт кодирования, дополнив его правилами, отражающими специфику данной ОС РВ и рассмотренными в предыдущих разделах.

Стандарт "Руководство по использованию языка Си для программного обеспечения встроенных систем" был разработан консорциумом MISRA для обеспечения дополнительной безопасности в программах, работающих в требовательных к безопасности системах. Следует, однако, заметить, что на настоящее время нет языка программирования, который может гарантировать, что программа будет делать то, что задумал программист. Существует большое число проблемных вопросов, которые могут возникнуть с любым языком программирования.

Язык Си является наиболее популярным языком программирования высокого уровня для встроенных систем. Однако в его стандарте существует несколько ситуаций, в которых поведение программы зависит от компилятора и/или компоновщика. Это обстоятельство делает язык Си неподходящим для использования при разработке систем, требующих переносимости, систем с повышенными требованиями безопасности, а также встроенных систем. Другой особенностью, которая приводит к проблемам, по мнению создателей документа, является то, что многие опечатки при написании кода могут не приводить к ошибкам при сборке программы. Например, написание " = " вместо " = = ". Подобная опечатка полностью меняет логику программы, но, согласно стандарту, это не является ошибкой.

Чтобы избежать, насколько это возможно, тех причин, которые могут привести к негативным последствиям, использование языка Си должно быть ограни-

чено. Именно эту цель и преследует данный стандарт. Он вводит некоторое подмножество языка Си, которое способствует написанию надежных программ. Стандарт MISRA-C содержит 127 правил, следование которым поможет существенно улучшить надежность и безопасность систем. Большая часть этих правил может проверяться непосредственно статическими анализаторами. Средства статического анализа в этом случае могут использоваться в процессе разработки для проверки правил принятого стандарта кодирования. В частности, для CSA может потребоваться дополнительное аннотирование исходного кода специфическими атрибутами, а для анализатора mygcc — создание набора правил, отражающих соответствующие положения стандарта кодирования.

### Заключение

Операционные системы реального времени предъявляют повышенные требования к качеству исходного программного кода, принуждая разработчиков интегрировать дополнительные средства в процесс разработки. Одним из таких средств может стать средство статического анализа, способное показать высокую эффективность при выявлении некоторых ошибок, которые характерны для разработчиков приложений для ОС РВ.

Одним из наиболее важных шагов для интеграции средств статического анализа является разработка стандарта кодирования. Стандарт кодирования является основой для автоматизации проверки качества кода. Проще всего этого добиться посредством интеграции средств статического анализа в систему управления конфигурациями и процессом сборки проекта, что позволит разработчикам проводить повторный анализ быстро и детерминировано.

В заключение следует отметить, что статический анализ не является гарантией решения всех проблем разработки ОС РВ. Однако это достаточно эффективный подход, и его применение может существенно помочь в повышении качества программного обеспечения. Следует, однако, отметить, что статический анализ является одним в числе многих других инструментариев, которые необходимо использовать для разработки качественного программного обеспечения. К их числу относятся, например, анализ требований, грамотное проектирование и тестирование.

### Список литературы

1. **Безруков В. Л.** и др. Введение в ос2000. — Вопросы кибернетики. М.: НИИСИ РАН, 1999.
2. **Годунов А. Н.** Операционная Система Реального Времени Багет 3.0 // Программные продукты и системы. 2010. № 4 (92). С. 15–19.
3. **Wind River Platform for Safety Critical ARINC 653.** URL: <http://www.windriver.com/products/product-overviews/Platform-Safety-Critical.pdf>
4. **ARINC.** ARINC Specification 653-2: Avionics Application Software Standard Interface Part 1. — Required Services. Maryland, USA: Aeronautical Radio INC, 2005.
5. **Clang Static Analyzer.** URL: <http://clang-analyzer.lvm.org>
6. **Проект mygcc.** URL: <http://mygcc.free.fr>
7. **Volanschi N.** Condate: A Proto-language at the Confluence Between Checking and Compiling // Proceedings of the Intl. Conf. on Principles and Practice of Declarative Programming (PPDP'06). Venice, Italy, 2006.
8. **MISRA-C.** Guidelines For The Use Of The C Language In Vehicle Based Software. URL: <http://www.misra-c.com>

Д. Ж. Корзун, канд. физ.-мат. наук, доц.,  
А. А. Ломов, аспирант,  
П. И. Ванга, аспирант,  
Петрозаводский государственный университет,  
e-mail: dkorzun@cs.karelia.ru

# Автоматизированная модельно-ориентированная разработка программных агентов для интеллектуальных пространств на платформе Smart-M3

Рассматривается задача автоматизации разработки приложений для платформы Smart-M3, когда структура предметной области задается OWL-онтологией. Программный агент определяет индивидуальную онтологию для взаимодействия в приложении на основе своей роли и интересов. По онтологии генерируется онтологическая библиотека на нужном языке программирования. Преобразование высокоуровневой модели OWL в низкоуровневую модель RDF скрыто от разработчика. Предлагаемое решение реализовано в программном инструментарии SmartSlog платформы Smart-M3.

**Ключевые слова:** интеллектуальные пространства (ИП), Smart-M3, OWL, RDF, онтологическая библиотека, кодогенерация

## Введение

Платформа Smart-M3<sup>1</sup> предназначена для создания многоагентных распределенных приложений, взаимодействующих в общем интеллектуальном пространстве (*smart space*) [1, 2]. Оно определяет разделяемое хранилище динамической информации для создания сред *повсеместных вычислений* (*ubiquitous computing*) [3, 4]. Приложение строится как набор процессоров знаний (*knowledge processor* — *KP*). Каждый *KP* — это программный агент. Агенты могут работать на разнородных вычислительных устройствах среды повсеместных вычислений (сенсоры, бытовая техника, мобильные устройства, серверные ЭВМ и др.) и взаимодействуют через сетевой доступ к интеллек-

туальному пространству. Разделяемая информация представлена в распределенном хранилище по модели представления *RDF* (*resource description framework*). Хранилище управляется семантическим информационным брокером (*semantic information broker* — *SIB*). Такое *RDF*-представление допускает семантический анализ: базовые примитивы доступны на стороне *SIB*, более сложные могут быть реализованы на стороне *KP*.

В данной статье рассматривается задача автоматизации модельно-ориентированной разработки *KP* для платформы Smart-M3. Решения для этой задачи реализуются в инструментарии SmartSlog<sup>2</sup>. Используется язык OWL (от англ. *web ontology language*) — стандарт представления знаний в семантическом веб [5]. Предлагается способ разработки приложений, в которых

<sup>1</sup> <http://sourceforge.net/projects/smart-m3/>

<sup>2</sup> <http://sourceforge.net/projects/smartslog/>

каждый *КР* использует OWL-онтологию предметной области, определяя требуемую ему структуру представления данных для взаимодействия с другими *КР* через ИП. Для реализации взаимодействия логика *КР* программируется в высокоуровневых онтологических терминах (класс, свойство, индивид) и коммуникационных примитивах (публикация, подписка, семантический запрос и др.). Низкоуровневые детали реализации взаимодействия скрыты от программиста. Вместе с тем, инструментарий SmartSlog учитывает разнообразие вычислительных устройств за счет поддержки различных языков программирования, автоматической генерации кросс-платформенного кода с малым числом зависимостей, экономного использования вычислительных и сетевых ресурсов и возможностей управления со стороны разработчика.

## Интеллектуальные пространства

Концепция повсеместных вычислений декларирует создание приложений, которые естественным и незаметным для человека образом автоматизируют решение задач повседневной жизни, предугадывая и адаптируясь под текущую ситуацию [6]. Такое приложение требует взаимодействия разнородных устройств, находящихся в окружении человека: от миниатюрных встроенных бытовых и мобильных устройств до персональных компьютеров, Интернет-серверов и систем облачных вычислений. На каждом устройстве работают программные агенты, реализующие распределенное приложение.

В качестве примера рассмотрим взаимодействие бытовых устройств в "умном доме". Агент на мобильном телефоне отслеживает начало и завершение звонка, информируя об этих событиях других агентов. Агент на музыкальном плеере автоматически уменьшает громкость на время телефонного разговора и возвращает предыдущее состояние по завершении.

Понятие ИП реализует концепцию повсеместных вычислений, определяя виртуальную и ориентированную на сервисы многопользовательскую информационную среду с динамическим взаимодействием [3, 4]. Наследуется асинхронная коммуникационная модель распределенных вычислений [7, 8], когда агенты взаимодействуют, используя информацию из общего пространства. Помимо базовых примитивов "сохранить" и "получить" (единовременные запросы), применяется модель публикации/подписки [9] для постоянных запросов. Они проактивно отслеживают изменения тех данных в ИП, на которые подписывается агент.

Как и в семантическом веб, информация в ИП представляется в виде утверждений (триплетов) [10], каждое состоит из субъекта, предиката и объекта. Триплеты определяются моделью *RDF* [11]. Субъект и объект соответствуют понятиям в предметной области, а предикат определяет отношение между ними. Имена для субъектов, предикатов и объектов определяются унифицированным идентификатором ресур-

сов (*uniform resource identifier* — URI). Такой подход позволяет использовать термин "знания" для информации в ИП. Помимо собственно данных, определяются семантические отношения между ними, что приводит к представлению в виде *RDF*-графа. Также модель *RDF* позволяет представить не только данные, но и вычисления [12]. Операции доступа к ИП сводятся к семантическим запросам, при их выполнении используются методы сопоставления для поиска в *RDF*-графе [13, 11]. Так, в примере "умный дом" возможен запрос: "получить список звуковых устройств, активных в настоящий момент".

Агенты в ИП могут быть ассоциированы с сервисами и предоставляют актуальные знания для использования другими агентами. Агенты, связанные с персональными устройствами пользователя, являются непосредственными потребителями этих знаний. Агенты могут публиковать в ИП контекстную информацию о пользователях и их окружении (текущие координаты, сердечный ритм, температуру среды и т. д.). Некоторые агенты анализируют текущие факты из ИП и выводят новые знания [3, 11], чтобы, например, автоматически инициировать запросы к сервисам. В общем случае ИП объединяет такие неоднородные информационные источники, как информация о физическом окружении с измерительных устройств (например датчики в доме или автомобиле), информация от окружающих сервисов (например Интернет-сервисы или сервисы туристических точек интереса), персональная информация о пользователях (например профайл или адресная книга).

Таким образом, приложению предоставляется абстрактная инфраструктура ИП, которая "позволяет участникам накапливать и применять знания о текущем окружении и адаптироваться для улучшения деятельности в этом окружении" [3]. Такая абстракция допускает построение информационной канонической модели и предметных посредников, что обеспечивает интероперабельность при интеграции неоднородных информационных ресурсов [14]. Предполагается высокий уровень масштабирования инфраструктуры, что позволяет рассматривать ИП как кандидата для реализации концепции "Интернет вещей" (IoT — *Internet of Things*) [15].

## Платформа Smart-M3

Платформа Smart-M3 реализует инфраструктуру для приложений в интеллектуальном пространстве и предоставляет инструментарий ADK (*application development kit*) для создания таких приложений. Аббревиатура M3 подчеркивает ориентацию на свойства multi-device, multi-vendor и multi-domain, требуемые концепцией IoT. Кратко представим возможности инфраструктуры для построения приложений, инструментарий будет рассмотрен в следующем разделе. Подробное описание, включающее принципы создания приложений и примеры развиваемых приложений, представлено в работах [16, 1, 2].

Ключевым элементом инфраструктуры — семантический информационный брокер *SIB*, являющийся точкой доступа к ИП. Каждый *SIB* управляет некоторой частью разделяемых знаний, которая представлена как набор *RDF*-триплетов (*RDF*-хранилище). Платформа Smart-M3 предоставляет механизмы рассуждений для получения новых знаний из имеющихся, что сводится к анализу *RDF*-графа для хранения набора триплетов [11, 13] (например, поддержка онтологического свойства *owl:sameAs* для определения, когда два идентификатора соответствуют одному понятию). Для этого используется язык семантических запросов, текущая реализация Smart-M3 поддерживает язык WilburQL, планируется переход на язык SPARQL [17].

Через *SIB* к ИП динамически подключаются разнообразные устройства и сервисы посредством программных агентов, называемых процессорами знаний *KP*, обеспечивая свойство multi-device. Доступ к разделяемым знаниям в ИП реализуется операциями вставки, удаления, обновления и подписки, которые *KP* запрашивает у *SIB* по протоколу SSAP (*Smart Space Access Protocol*). Последний может работать поверх таких протоколов, как HTTP, NoTA, Bluetooth, которые поддерживаются большинством производителей устройств, обеспечивая свойство multi-vendor. Каждый *KP* получает и размещает в ИП знания в соответствии с моделью *RDF* без обязательной привязки к какой-либо онтологии и может использовать собственную интерпретацию имеющихся знаний и обработку их возможной несогласованности. Такой подход обеспечивает свойство multi-domain, когда информация поступает от источников из различных предметных областей и интероперабельность поддерживается *RDF*-связями.

Схема инфраструктуры и использующих ее многоагентных приложений представлена на рис. 1. В общем случае ИП может быть реализовано с помощью

нескольких *SIB*, объединенных в сеть и поддерживающих виртуальную интеграцию данных внутри одного ИП. Доступ к ИП выполняется через произвольный *SIB* этой сети. Приложение создается как динамический набор *KP*, каждый выполняет определенный сценарий, интерпретируя некоторую разделяемую с другими *KP* часть ИП. Сценарий управляется наблюдаемыми изменениями в ИП вследствие действий других *KP*. Некоторые сценарии являются транзитными — их выполнение изменяется в зависимости от прихода и выхода других *KP* в ИП (например, при изменении местоположения изменяется состав окружающих сервисов). Интегрированные приложения используют несколько ИП — в приложение входят *KP*, имеющие одновременный доступ к этим ИП [18].

Отметим, что платформа Smart-M3 предоставляет весьма большие возможности по созданию приложений по сравнению с аналогами. Например, платформа Information Workbench [19] для приложений семантического веб выделяет такие платформенные концепты как "провайдер данных", "коннектор" и "интерфейс отображения данных", интегрирующие данные (из внутренних и внешних источников) в центральном хранилище. Концепты предоставляет сама платформа, а приложение лишь использует интегрированные данные.

### Разработка Smart-M3 приложений на основе OWL-онтологии предметной области

В качестве базового системного ПО для создания приложений платформа Smart-M3 определяет *KP*-интерфейс *KPI* (*KP interface*), реализуемый для конкретного языка программирования и вычислительной среды. Доступный на данный момент набор *KPI* представлен в табл. 1. Разработчик, программирующий логику *KP*, вызывает функции *KPI* для выполнения операций доступа к ИП. Низкоуровневый вариант разработки использует *RDF*-триплеты как базовые параметры этих операций. Соответствующий *KPI* непосредственно реализует клиентскую часть протокола SSAP, а логика *KP* должна интерпретировать и обрабатывать на уровне триплетов знания предметной области.

Практически значимые предметные области приводят к большим наборам триплетов. Для автоматической обработки такие объемы могут не являться критичными, но использование *RDF*-модели в алгоритмах логики *KP* и программной реализации приводит к громоздкому и труднообозримому программному коду. Более выгодно использовать подход на основе высокоуровневых онтологических моделей представления данных. Он реализуется авторами в инструментарии SmartSlog. Последний в настоящее время становится основным высокоуровневым ADK платформы Smart-M3.

Под онтологией будем понимать формальное определение понятий и отношений между ними с помо-



Рис. 1. Инфраструктура интеллектуальных пространств в виде сетей из семантических информационных брокеров (*SIB*), предоставляемая платформой Smart-M3. Многоагентные приложения состоят из агентов — процессоров знаний *KP*. Каждый *KP* работает на некотором вычислительном устройстве и имеет доступ к одному или более ИП

Доступные реализации *KPI* для платформы Smart-M3

№ п/п	<i>KPI</i>	Средства программирования <i>KP</i>	Автор	Публичный репозиторий
<b>Низкоуровневые <i>KPI</i> (операции с <i>RDF</i>-триплетами)</b>				
1	Whiteboard, Whiteboard-Qt	C/Glib, C/DBus, C++/Qt	Исследовательский центр Nokia (Хельсинки, Финляндия)	<a href="http://sourceforge.net/projects/smart-m3/">http://sourceforge.net/projects/smart-m3/</a>
2	M3-Python <i>KPI</i> (m3 kp)	Python	Исследовательский центр Nokia (Хельсинки, Финляндия)	<a href="http://sourceforge.net/projects/smart-m3/">http://sourceforge.net/projects/smart-m3/</a>
3	<i>KPI_Low</i>	ANSI C (ограниченный набор для программирования малопроизводительных устройств)	Научно-технический Центр Финляндии VTT (Оулу, Финляндия)	<a href="http://sourceforge.net/projects/kpilow/">http://sourceforge.net/projects/kpilow/</a>
4	Smart-M3 Java <i>KPI</i> library	Java ( <i>KP</i> работает через веб-обозреватель)	Болонский университет (Болонья, Италия) и Научно-технический Центр Финляндии VTT (Оулу, Финляндия)	<a href="http://sourceforge.net/projects/smartm3-javakpi/">http://sourceforge.net/projects/smartm3-javakpi/</a>
5	Smart-M3 PHP <i>KPI</i> library	PHP ( <i>KP</i> работает через веб-обозреватель)	Болонский университет (Болонья, Италия)	<a href="http://sourceforge.net/projects/sm3-php-kpi-lib/">http://sourceforge.net/projects/sm3-php-kpi-lib/</a>
6	C# <i>KPI</i> for Smart-M3	.NET, C#	Болонский университет (Болонья, Италия)	<a href="http://sourceforge.net/projects/m3-csharp-kpi/">http://sourceforge.net/projects/m3-csharp-kpi/</a>
<b>Высокоуровневые <i>KPI</i> (операции с онтологическими объектами)</b>				
7	Smart-M3 OWL-DL to C(glib) API Generator	C/Glib, C/DBus	Университет Åbo Akademi (Турку, Финляндия)	<a href="http://sourceforge.net/projects/smart-m3/">http://sourceforge.net/projects/smart-m3/</a>
8	Smart-M3 Ontology to Python API Generator	Python	Исследовательский центр Nokia (Хельсинки, Финляндия)	<a href="http://sourceforge.net/projects/smart-m3/">http://sourceforge.net/projects/smart-m3/</a>
9	SmartSlog	ANSI C (включая ограничения малопроизводительных устройств); .NET, C#	Петрозаводский государственный университет (Петрозаводск, Россия)	<a href="http://sourceforge.net/projects/smartslog/">http://sourceforge.net/projects/smartslog/</a>

стью логики предикатов первого порядка [20]. Понятия и отношения отображаются в элементы данных из различных неоднородных источников предметной области приложения. Основными элементами являются: *класс* (соответствует понятию из предметной области, например человек или телефонный звонок), *индивид* (конкретный объект, например человек Иванов И. И.), *свойство* (отношение между элементами данных и понятиями, например имя человека или выполняемый им сейчас звонок) и *значение свойства* (элемент данных или отношение с конкретным объектом, например строка "Иванов" или ссылка на индивида "звонок"). Классы и свойства определяют структуру предметной области (граф классов), а индивиды и значения свойств (граф индивидов) — конкретный набор фактов и связей между ними.

Для спецификации подобных онтологий можно использовать язык OWL [5]. Помимо синтаксиса, для описания понятий и отношений предметной области предоставляется дополнительный набор средств моделирования предметной области, например наследование и пересечение классов, кардинальность, область значений свойств. *RDF*-граф допускает агрегированную интерпретацию в виде OWL-графа индивидов —

индивид представляется набором *RDF*-триплетов, и операции с индивидом и его свойствами сводятся к операциям с наборами триплетов.

OWL-онтология предметной области более удобна для разработчика *KP*, чем использование *RDF*-представления. Это свойство применяется в инструментарии SmartSlog для упрощения процесса разработки *KP* (рис. 2), что позволяет воспользоваться преимуществами онтологического модельно-ориентированного подхода к разработке ПО [21]. С помощью кодогенератора SmartSlog генерируется *онтологическая библиотека* по заданной OWL-онтологии. В результате разработчик может создавать программный код в терминах предметной области (т. е. в терминах заданной OWL-онтологии). Программный код логики *KP* использует переменные для индивидов и их свойств и вызывает коммуникационные примитивы взаимодействия с ИП. Детали, связанные с низкоуровневой обработкой триплетов и операциями протокола SSAP, скрыты от разработчика.

Для доступа к ИП SmartSlog использует низкоуровневый *KPI*, который реализует непосредственное взаимодействие с ИП по протоколу SSAP. Текущая версия SmartSlog использует библиотеку *KPI\_low* (см.

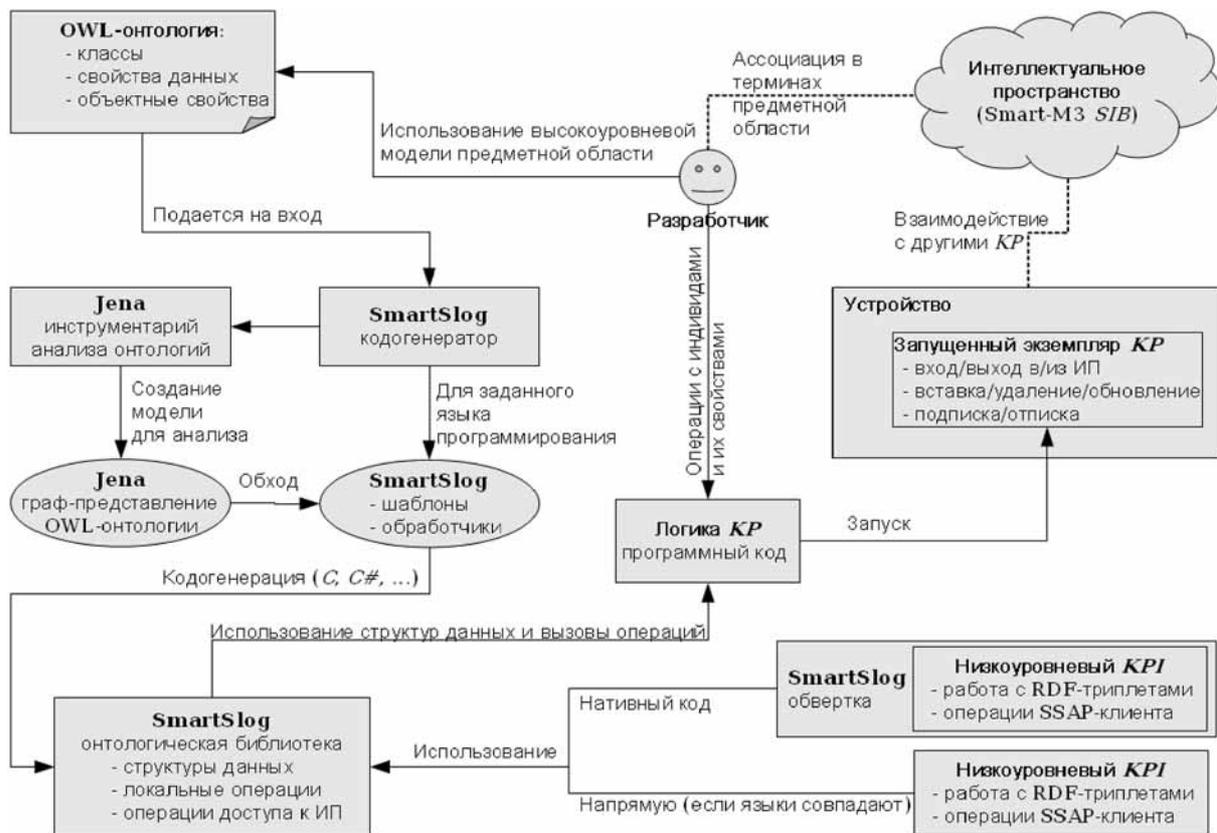


Рис. 2. Общая схема использования инструментария SmartSlog для автоматизированной разработки КР

строку 3 в табл. 1). Она ориентирована на малопродуктивные устройства (встраиваемые устройства, бытовая техника, мобильные компьютеры и т. п.). Число используемых зависимостей от других библиотек минимизировано. Исходный код ограничен подмножеством языка ANSI C. При использовании онтологической библиотеки SmartSlog в программе на языке C вызов *KPI\_low* осуществляется напрямую. В случае языка C# применяется обертка для вызовов *KPI\_low* (нативный код). Для подключения другого низкоуровневого *KPI* достаточно создать соответствующую обертку.

### Генерация онтологической библиотеки

Структура предметной области задается на языке OWL. Соответствующая OWL-онтология предоставляется разработчику *KP* готовой или он создает ее из предоставленных онтологий, отбирая необходимые понятия и отношения и добавляя нужные отношения. По OWL-онтологии генерируется программный код методами автоматического программирования [22], реализующий структуры данных для представления в программном коде *KP* объектов предметной области. Отображение OWL-онтологии в программный код основано на описательной близости онтологического и

объектно-ориентированного (ОО) подходов. На рис. 3 показаны схематичные примеры для языков C и C#. Первый является процедурным, и OWL-классы реализуются стандартными C-структурами. Второй — объектно-ориентированный, и OWL-классы реализуются в виде ОО-иерархии классов.

Кодогенератор SmartSlog использует статическую схему с независимыми от входной онтологии шаблонами и обработчиками. Шаблоны представляют собой части кода, через которые реализуется описание онтологических классов и свойств. Шаблоны определены с точностью до имени используемых в коде элементов конкретной онтологии. Шаблон содержит специальные теги `<name>` вместо реальных имен. Обработчик осуществляет преобразование одного или нескольких шаблонов в программный код, заменяя теги именами, взятыми из онтологии. Онтология представляется как граф с помощью инструментария для анализа онтологий Jena<sup>3</sup>. Кодогенератор обходит вершины построенного графа, вызывая соответствующие обработчики.

Такая схема реализует горизонтальную трансформацию, поскольку затрагиваются только имена струк-

<sup>3</sup> <http://jena.sourceforge.net/>

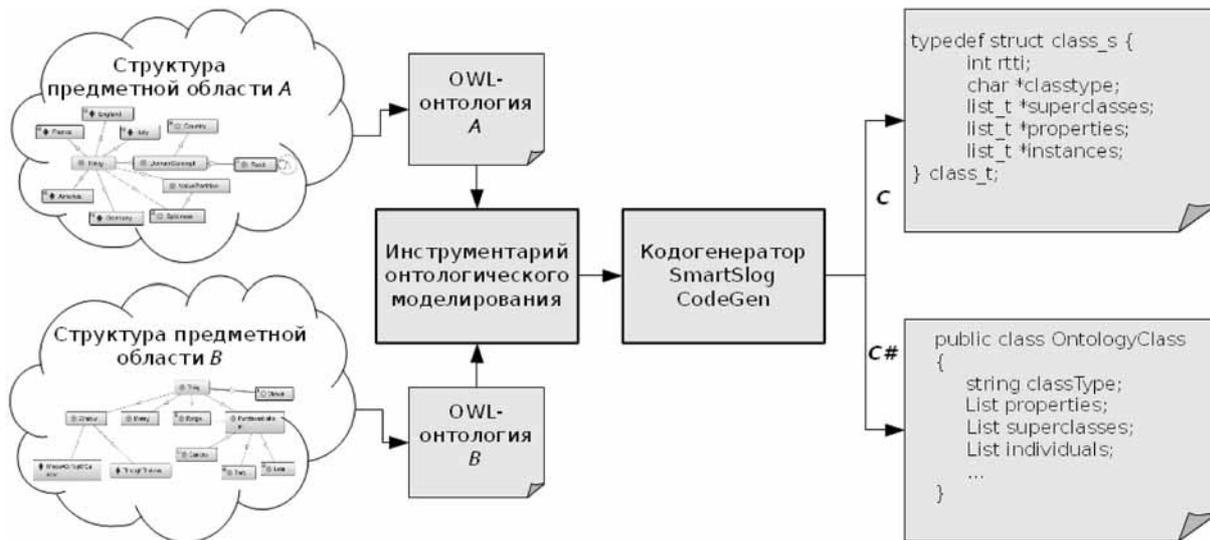


Рис. 3. Генерация кода структур данных (на требуемом языке программирования), которые затем использует разработчик для представления знаний в соответствии с заданной онтологией

тур данных и аргументы методов [22]. Для каждого поддерживаемого языка программирования необходимо создавать свой набор шаблонов и обработчиков. Вследствие специфики языка программирования и вычислительной среды может потребоваться модификация алгоритма обхода графа онтологии.

Сходные способы кодогенерации используются и в других задачах онтологического подхода к разработке ПО, но, в основном, концентрируясь на ОО-языках (в первую очередь, Java) или даже на интерпретируемых ОО-языках (например Python, Ruby) [21]. В случае процедурных компилируемых языков необходимо более сложно реализуемый контроль семантической корректности во время выполнения программы.

Для Smart-M3 ИП характерно, что его части, даже пересекающиеся, могут быть структурированы при помощи разных онтологий. Каждый КР может использовать индивидуальную онтологию для интерпретации доступных ему знаний. Для разработки желательны средства построения индивидуальной онтологии на основе различных композиций базовых онтологий. Интеграция, слияние, отображение онтологий и другие операции порождают класс задач, связанных с созданием КР, работающих одновременно с неоднородными источниками знаний или управляющих состоянием других участников [2]. Для таких целей необходимо использовать инструментарий онтологического моделирования и преобразования. Например, для задачи объединения онтологий можно использовать инструментарий Protégé<sup>4</sup> (см. рис. 3).

<sup>4</sup> <http://protege.stanford.edu/>

### Интерфейс прикладного программирования (API) онтологической библиотеки SmartSlog

Онтологическая библиотека состоит из зависимой и независимой от онтологии частей (рис. 4). Первая формируется сгенерированным кодом. Независимая от онтологии часть реализует операции над объектами, включая локальную обработку на стороне КР и взаимодействие с разделяемым ИП. Предоставляется интерфейс прикладного программирования (API — *Application Programming Interface*), разработчик указывает соответствующие вызовы API-функций, программируя логику КР. Независимость от онтологии обеспечивается тем, что имена функций и типы аргументов не привязаны к конкретным именам из заданной онтологии. Объекты передаются как параметры, тип которых конкретизируется лишь до общеонтологического уровня "класс", "индивид", "свойство".

Помимо SmartSlog, такой подход используется в КР<sub>И</sub>, представленном в строке 7 табл. 1. В этом случае API библиотеки становится "универсальным", в отличие, например, от API высокоуровневого КР<sub>И</sub>, представленного в строке 8 табл. 1, в котором имена понятий и отношений онтологии фиксируются в именах генерируемых функций. В то же время универсальность требует выполнения проверки передаваемых параметров во время выполнения программы.

Деление библиотеки на две части повышает эффективность разработки приложений. Независимая часть может быть использована любым КР, независимо от его индивидуальной онтологии, а также не требуется перекомпиляция при внесении изменений в онтологию. Зависимая часть может быть использована несколькими КР, если они используют одну он-

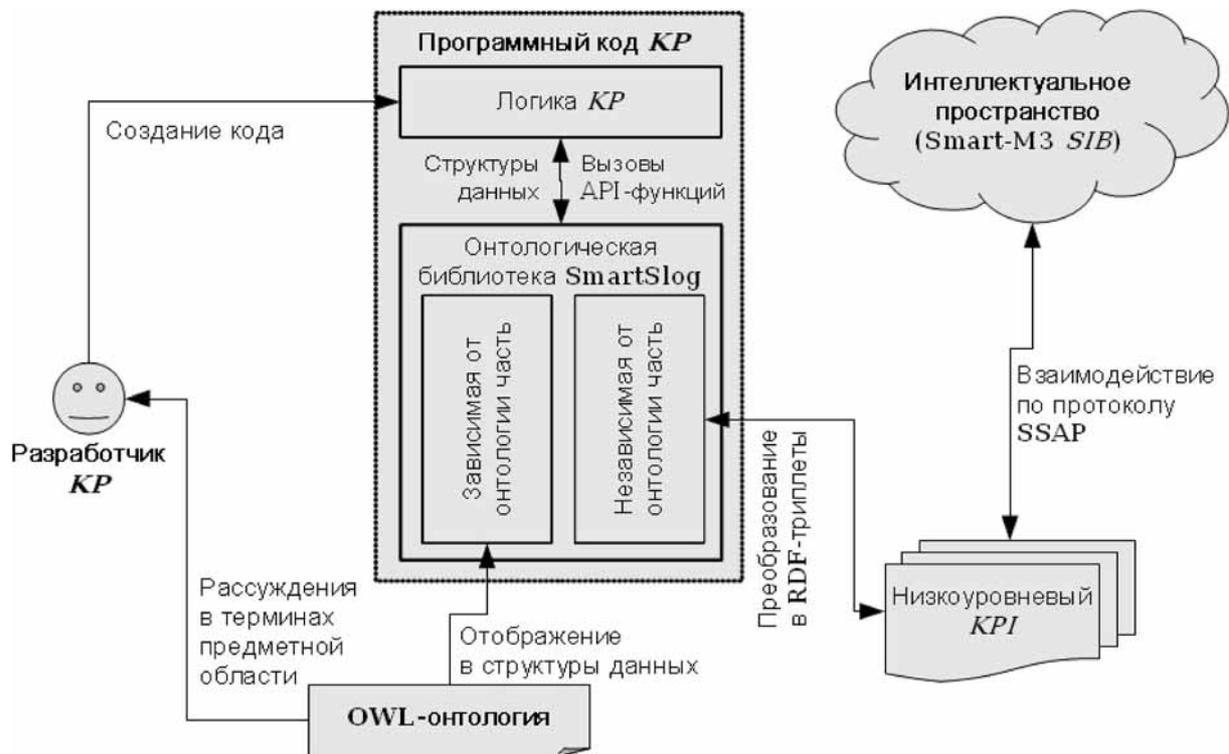


Рис. 4. Структура и схема использования онтологической библиотеки SmartSlog

тологию. Такая ситуация нередко возникает в рамках как одного приложения, так и различных приложений для одной предметной области.

API библиотеки предоставляет две основные группы функций: операции с локальными онтологическими объектами и операции доступа к ИП. Поддерживается работа с ИП на уровне сессий: примитивы входа (*join*) и выхода (*leave*) с параметрами, определяющими SIB (имя, IP-адрес и порт). Примеры использования библиотеки SmartSlog представлены в табл. 2 для варианта библиотеки на языке С. Локальные операции позволяют создавать и удалять индивидов требуемого класса, устанавливать и изменять их свойства без отражения в ИП. При выполнении операций реализована проверка на корректность (например, кардинальность и область возможных значений свойства). Функции группы операций доступа к ИП реализуют стандартные примитивы на уровне индивидов и их свойств: вставка (*insert*), удаление (*remove*), обновление (*update*) в ИП и получения (*get*) из ИП по семантическому запросу. Синтаксически имена функций доступа выделяются префиксом "ss\_".

Наиболее важными в этой группе функций доступа выступают функции для операции подписки (*subscribe*). При инициализации подписки определяется набор индивидов и соответствующих им свойств, изменения которых в ИП должны отслеживаться. В синхронном варианте подписки КР блокируется

при вызове функции подписки, ожидая обновлений. Такой вариант можно использовать, даже если устройство не поддерживает многопоточность. Асинхронный вариант реализуется с помощью отдельного потока (свой для каждой подписки), который контролирует обновления из пространства и синхронизирует значения локальных переменных с ИП. Обновления приходят в фоновом режиме, т. е. КР работает в режиме "работа с последней версией данных".

Особенностью SmartSlog являются "шаблоны знаний", которые разработчик может использовать для поиска онтологических объектов в ИП и для локального отбора отправляемых в ИП данных. Каждый шаблон можно представить как абстрактного индивида (или граф абстрактных индивидов). Для такого индивида указываются только те свойства, которые нужны в операции, причем для каждого такого свойства указывается флаг *checked* или *unchecked*. Если свойство помечено флагом *checked*, то указывается требуемое значение свойства (например, отбор индивидов со свойством, равным указанному значению). Если свойство помечено флагом *unchecked*, то такого свойства не должно быть у искомого индивида (например, поиск всех индивидов класса "мужчина", у которого не установлено свойство "жена"). В дальнейшем возможно расширение шаблонов знаний за счет добавления условий на значения свойств (например больше, меньше).

Примеры использования онтологической библиотеки SmartSlog в программном коде КР

Описание действий	Пример программного кода
Получение доступа к ИП (открытие сессии). Нужная сессия определяется идентификатором ИП "X", IP-адресом и портом <i>SIB</i> . Также инициализируются структуры данных для локального хранения онтологических объектов	<pre>init_ss_with_parameters("X", "194.85.173.9", 10010); register_ontology();  if (ss_join(get_ss_info(), "KP_2") == -1) {     printf("Can't join to SS\n");     return 0; }</pre>
Создание индивида класса "женщина" и установка свойства "фамилия"	<pre>individual_t *aino = new individual(CLASS_WOMAN); set_property(aino, PROPERTY_LNAME, "Peterson");</pre>
Вставка индивида <i>aino</i> в ИП с последующим изменением в ИП значения его свойства "фамилия". Далее запрашивается из ИП текущее значение координат для индивида <i>aino</i>	<pre>ss_insert_individual(aino); ... ss_update_property(aino, PROPERTY_LNAME, "Ericsson"); ... ss_get_property(aino, PROPERTY_COORD);</pre>
Шаблон знаний для обновления свойства "фамилия". Создается шаблон <i>aino_p</i> , и операция обновления в ИП затронет только выбранное свойство	<pre>pattern_t *aino_p = new pattern(CLASS_WOMAN); add_check_prop_to_pattern(aino_p, PROPERTY_LNAME, NULL); ss_update_by_pattern(aino, aino_p);</pre>
Шаблон знаний для поиска в ИП всех мужчин "Тимо", у каждого жену зовут "Аино". Создаются два шаблона (для "Тимо" и "Аино") с соответствующими свойствами ("фамилия" и "жена"). Результат из ИП — набор всех найденных индивидов класса "мужчина"	<pre>pattern_t *timo_p = new pattern(CLASS_MAN); pattern_t *aino_p = new pattern(CLASS_WOMAN); add_check_prop_to_pattern(timo_p, PROPERTY_FNAME, "Тимо"); add_check_prop_to_pattern(aino_p, PROPERTY_FNAME, "Аино"); add_check_prop_to_pattern(timo_p, PROPERTY_HAS_WIFE, aino_p); timo_list = ss_get_individuals_by_pattern(timo_p);</pre>
Синхронная подписка (параметр <i>false</i> при открытии). В контейнер помещается набор индивидов и их свойств ( <i>timo</i> — PROPERTY_DRINK). После открытия подписки КР выполняет проверки изменения данных функцией <i>wait_subscribe()</i> , в блокируемом режиме, ожидая поступления обновлений для подписанных свойств индивидов	<pre>subscription_container_t *container = new subscription_container(); list_t *properties = list_get_new_list(); list_add_data(PROPERTY_DRINK, properties);  add_individual_to_subscription_container(container, timo, properties); if (ss_subscribe_container(container, false) != 0) { return 0; } while (true) {     prop_val_t *p_val2 = get_property(timo, PROPERTY_DRINK-&gt;name);     wait_subscribe(container); } ss_unsubscribe_container(container);</pre>
Асинхронная подписка (параметр <i>true</i> при открытии). Не требуется вызова дополнительных функций для проверки подписки. Обновления приходят в фоновом режиме параллельно с работой логики КР	<pre>if (ss_subscribe_container(container, true) != 0) { return 0; } while (true) {     prop_val_t *p_val2 = get_property(timo, PROPERTY_DRINK-&gt;name);     ... } ss_unsubscribe_container(container);</pre>

### Дополнительные возможности

Онтологическая библиотека скрывает от разработчика двунаправленное преобразование высокоуровневых онтологических объектов в триплеты. Любой вызов функции для взаимодействия с ИП требует выполнения такого преобразования для входных онтологических объектов и последующий вызов низкоуровневого *KPI* с передачей ему соответствующих триплетов и выполнения запросов к ИП по протоколу *SSAP*. Ответный результат представляет собой набор триплетов, поэтому необходимо их преобразование в соответствующие онтологические объекты.

Отметим, что преобразование индивидов и их свойств в *RDF*-представление реализуется очевидным

образом. Обратное преобразование не всегда возможно, что вызвано следующими двумя причинами.

- Низкоуровневое представление в виде набора триплетов не гарантирует существование соответствия триплета индивиду. Для указания соответствия значений свойств необходимо при публикации данных добавлять триплет с предикатом *rdf:type*. Он задает связь между классом и идентификатором индивида. Идентификатор далее используется в триплетах для связывания значений свойств с индивидом.

- Произвольная вставка триплетов в ИП может привести к семантической некорректности. Например, когда свойство "друг" индивида класса "человек" будет ссылаться на индивида класса "овощ".

Платформа Smart-M3 оставляет контроль этих причин на стороне приложения, т. е. каждый *KP* сам должен интерпретировать имеющиеся в ИП знания. При использовании SmartSlog выполняются проверки, не допускающие возникновения подобных ситуаций по вине разрабатываемого *KP*. В то же время работа других *KP* может приводить к этим ситуациям, но библиотека SmartSlog не предоставит доступ к содержимому, не соответствующему используемой онтологии. При необходимости разработчику придется использовать собственный низкоуровневый код для работы с таким содержимым.

Инструментарий SmartSlog ориентирован на разработку приложений для широкого спектра устройств, включая встроенные и низкопроизводительные. Онтологическая библиотека использует ограниченный набор сторонних компонентов, уменьшая зависимость от вычислительной платформы. При работе *KP* не создается локального хранилища *RDF*-триплетов: преобразование между онтологическими объектами и триплетами происходит непосредственно при выполнении операций доступа к ИП, локальная память освобождается сразу после использования триплетов. В инструментарии SmartSlog предопределены специальные константы для выбора тех элементов из онтологии, которые нужны в коде *KP* (или для исключения неиспользуемых), что предоставляет разработчику способ управлять объемом требуемой памяти.

### Заключение

Представленный инструментарий SmartSlog выступает ключевым высокоуровневым ADK для платформы Smart-M3. Используется онтологический подход к разработке ПО для таких парадигм распределенных приложений будущего, как "Интернет вещей" и "повсеместные вычисления". Определена схема использования инструментария для автоматизации разработки приложений. Описаны основные возможности инструментария и предложены способы их реализации на основе низкоуровневых примитивов платформы Smart-M3. За счет ограничений на код программной реализации онтологической библиотеки поддержки многоязычности учитывается современное разнообразие вычислительных устройств и сред. Шаблоны знаний и операция подписки реализуют эффективные способы взаимодействия с ИП, позволяющие избежать излишней передачи и обработки данных. В результате разработчик получает, с одной стороны, высокоуровневый инструмент проблемно-ориентированной разработки, а с другой стороны, имеются достаточные средства для обеспечения совместной работы большого числа разнообразных вычислительных устройств в рамках динамически формируемых многоагентных приложений.

### Список литературы

1. **Honkola J., Laine H., Brown R., Tyrkkö O.** Smart-M3 Information Sharing Platform // Proc. of IEEE Symp. Computers and Communications (ISCC'10), 2010. P. 1041–1046.
2. **Korzun D. G., Balandin S. I., Luukkala V., Liuha P., Gurtov A. V.** Overview of Smart-M3 Principles for Application Development // Сб. тр. Межд. конгресса по интеллектуальным системам и информационным технологиям (IS & IT'11), конференция "Интеллектуальные системы" (AIS'11), 2011. С. 64–71.
3. **Cook D. J., Das S. K.** How smart are our environments? An updated look at the state of the art // Pervasive and Mobile Computing. 2007. Vol. 3, N 2. P. 53–73.
4. **Oliver I., Honkola J., Ziegler J.** Dynamic, localized space based semantic webs // Proc. of IADIS Int'l Conf. WWW/Internet, 2008. P. 426–431.
5. **OWL Web Ontology Language Reference.** W3C Recommendation. 2004. URL: <http://www.w3.org/TR/owl-ref/>
6. **Weiser M.** The Computer for the Twenty-First Century // Scientific American. 1991. Vol. 265. N 3. P. 94–10.
7. **Gelernter D.** Generative communication in Linda // ACM Transactions on Programming Languages and Systems. 1985. Vol. 7. N 1. P. 80–112.
8. **Fensel D.** Triple-space computing: Semantic web services based on persistent publication of information // Proc. of IFIP Int'l Conf. Intelligence in Communication Systems (INTELLCOMM 2004). LNCS 3283. Springer, 2004. P. 43–53.
9. **Eugster P. T., Felber P. A., Guerraoui R., Kermarrec A.-M.** The many faces of publish/subscribe // ACM Computing Survey. 2003. Vol. 35. N 2. P. 114–131.
10. **Nixon L. J. B., Simperl E., Krummenacher R., Martínrecuerda F.** Tuplespace-based computing for the Semantic Web: A survey of the state-of-the-art // The Knowledge Engineering Review. 2008. Vol. 23. N 2. P. 181–212.
11. **Gutierrez C., Hurtado C. A., Mendelzon A. O., Perez J.** Foundations of Semantic Web databases // Journal of Computer and System Sciences. 2011. Vol. 77. N 3. P. 520–541.
12. **Rodriguez M. A.** General-Purpose Computing on a Semantic Network Substrate // Emergent Web Intelligence: Advanced Semantic Technologies. Series "Advanced Information and Knowledge Processing" / R. Chbeir, A. Hassani, A. Abraham, Y. Badr. (Eds.). Springer-Verlag, 2010. P. 57–104.
13. **Euzenat J., Shvaiko P.** Ontology matching. Springer-Verlag, 2007. 341 p.
14. **Калиниченко Л. А.** Методология организации решения задач над множественными распределенными неоднородными источниками информации // Сб. тр. Межд. конф. "Современные информационные технологии и ИТ-образование". М.: Изд-во МГУ, 2005. С. 20–37.
15. **Gómez-Goiri A., López-de-Ipiña D.** A Triple Space-Based Semantic Distributed Middleware for Internet of Things // Proc. of 10th Int'l Conf. Current trends in web engineering (ICWE 2010). LNCS 6385. Springer-Verlag, 2010. P. 447–458.
16. **Balandin S. I., Waris H.** Key properties in the development of smart spaces // Proc. of 5th Int'l Conf. Universal Access in Human-Computer Interaction (UAHCI'09). Part II: Intelligent and Ubiquitous Interaction Environments. Springer-Verlag, 2009. P. 3–12.
17. **SPARQL Query Language for RDF / W3C Recommendation.** Prud'hommeaux E., A. Seaborne (Eds.). 2008. URL: <http://www.w3.org/TR/rdf-sparql-query/>
18. **Korzun D. G., Galov I. V., Kashevnik A. M., Shilov N. G., Krinkin K. V., Korolev Y.** Integration of Smart-M3 Applications: Blogging in Smart Conference // Proc. of 11th Int'l Conf. Next generation wired/wireless networking and 4th Int'l Conf. Smart spaces (NEW2AN'11/ruSMART'11). LNCS 6869. Springer, 2011. P. 51–62.
19. **Госсен А., Хаазе П., Шмидт М.** Разработка Linked Data приложений на примере The Information Workbench // Сб. тр. конф. "Инженерия знаний и технологии семантического веба — 2011". СПб.: НИУ ИТМО, 2011. С. 19–26.
20. **Guarino N.** Formal ontology, conceptual analysis and knowledge representation // International Journal of Human-Computer Studies. Special issue: the role of formal ontology in the information technology. 1995. Vol. 43. N 5–6. P. 625–640.
21. **Разумовский А. Г., Пантелеев М. Г.** Использование онтологий в разработке программного обеспечения // Сб. тр. конф. "Инженерия знаний и технологии семантического веба — 2011". СПб.: НИУ ИТМО, 2011. С. 88–95.
22. **Czarnecki K., Eisenecker U. W.** Generative Programming: Methods, Tools, and Applications. Addison-Wesley, 2000. 832 p.

**В. В. Костюк**, канд. техн. наук, проф.,  
**М. А. Бовкунович**, студент,  
Российский государственный университет инновационных технологий  
и предпринимательства,  
e-mail: Viacheslav.Kostyuk@itbu.ru

## Элементы сервис-ориентированной разработки архитектуры прикладного программного обеспечения на основе бизнес-моделей

*Рассматриваются вопросы получения интерфейсов Web-сервисов на основе принципа "сверху-вниз", т. е. от моделей к коду путем трансформаций моделей, начиная с бизнес-уровня и до уровня кода. Главная цель рассмотренных трансформаций — решить проблему декомпозиции информационной системы, соответствующей бизнес-процессам. Эта цель достигается, что отвечает основной концепции SOA — слияние бизнес-логики с логикой информационной технологии, реализующей эту бизнес-логику.*

**Ключевые слова:** бизнес-моделирование, сервис-ориентированная архитектура, SOA, трансформации, генерация кода, WSDL

С каждым годом все большее внимание разработчиков программного обеспечения привлекают к себе средства моделирования программного кода. Этапы развития этих средств:

- ручное моделирование (например, посредством блок-схем на командном и на процедурном уровнях с ручным кодированием процедур);
- моделирование средствами Case-технологий с ручным написанием алгоритмов с последующей компиляцией кодов (например, программный продукт BPwin, реализующий стандарты серии IDEF);
- моделирование средствами Case-технологий с генерацией каркасных кодов на основе созданных моделей (например, Rational Rose, стандарт UML).

Последний пример интересен тем, что в нем применяется метод трансформации одного уровня модели (абстракции) в другой. Здесь можно наблюдать два уровня трансформаций. Можно сначала создать модель на уровне диаграммы последовательности и выполнить трансформацию ее в диаграмму классов, осуществив генерацию классов с методами. Затем построить диаграмму классов, добавив в каждый класс свойства, выполнить трансформацию этой диаграммы,

получив при этом каркасный код (алгоритм) на выбранном языке программирования.

Можно сразу создать модель на уровне диаграммы классов и выполнить в этом случае одноуровневую трансформацию этой модели в каркасный код. Этот вариант обычно используется при ясном видении декомпозиции разрабатываемого программного обеспечения.

При отсутствии такого видения необходима разработка моделей, включая диаграммы прецедентов, последовательности, кооперации, деятельности и, возможно, состояний. Во всем этом цикле разработки только две ранее упомянутые трансформации ориентированы на генерацию кода — это диаграмма последовательности, или диаграмма кооперации, и диаграмма классов. Трансформацию диаграммы последовательности в диаграмму кооперации надо расценивать как интерпретацию. Она ориентирована на генерацию кода как альтернатива диаграмме последовательности. Эти две диаграммы взаимно обратимы.

В качестве примера трансформации типа интерпретации можно привести понятия прямого и обратного проектирования, т. е. на основе модели можно получить код и, наоборот, на основе готового кода по-

лучить модель. Такая трансформация в среде Rational Rose пока еще не является совершенной, но как метод взаимной интерпретации хорошо представляется. Наиболее совершенными можно считать трансформации прямого и обратного проектирования при разработке баз данных. Модели баз данных в среде, например, Erwin или Rational Data Architect хорошо трансформируются в DDL (SQL)-код и наоборот.

Метод трансформаций получил широкое признание и развитие в среде разработки моделей и программного обеспечения на основе моделей в продукте IBM Rational Software Architect (RSA).

Прежде всего рассмотрим применение метода трансформаций типа интерпретации. Если ранее упомянутая такая трансформация представлялась в понятиях прямого и обратного проектирования, то в RSA наблюдается применение этого метода в понятиях так называемых принципов "снизу-вверх" (Bottom-Up), "сверху-вниз" (Top-Down) и "от краев к центру" [1].

Так, например, при разработке Enterprise JavaBean (EJB) на стадии установления соответствия с реляционной базой данных, которая предназначена для сохранения состояния сущностного компонента, установление соответствия методом "сверху-вниз" означает, что связи будут устанавливаться по сущностному компоненту, а таблица базы данных может быть создана позже. Установление соответствия методом "снизу-вверх" выполняется в точности наоборот: таблица базы данных уже существует и соответствие устанавливается по схеме таблицы. После этого можно по этим соответствиям сгенерировать сущностный компонент. Установление соответствия по принципу "от краев к центру" используется, если и таблица базы данных, и сущностный компонент уже существуют.

При разработке сервис-ориентированной архитектуры могут быть использованы два принципа интерпретаций: "снизу-вверх" и "сверху-вниз". "Снизу-вверх" — это от Java кода к Web-сервису, когда можно создать Web-сервис из существующего Java-компонента или сессионного EJB без состояния. "Сверху-вниз" — это от WSDL-документа к каркасу Web-сервиса, когда можно создать для Web-сервиса базовый Java-компонент или сессионный EJB без состояния по документу WSDL. После этого пользователь может изменять базовый, каркасный код в соответствии со своими нуждами.

Как и следовало ожидать, процесс совершенствования Case-технологий фирмой IBM Rational в направлении снятия проблем с генерацией WSDL-файлов продолжается и находит отражение в продуктах WebSphere Business Modeler и RSA [2].

Одна из сложнейших проблем определения архитектуры программного обеспечения методом декомпозиции информационных систем была решена на одном из самых высоких уровней абстракции, на уровне бизнес-моделирования. Разрабатывая модель бизнес-процессов, бизнес-аналитик, не задумываясь о сервисной ориентации системы, уже определяет компоненты системы на основе процессов, работ,

функций, определяет контракты взаимодействия между компонентами на основе бизнес-логики взаимосвязей между процессами и внутри процессов. На основе этих артефактов, путем трансформаций бизнес-модель дает возможность осуществлять генерацию WSDL-файлов, необходимых для создания сервис-ориентированных архитектур информационных систем.

На примере информационной системы добровольного медицинского страхования рассмотрим, как и какие трансформации проходит бизнес-модель в целях получения WSDL-файлов.

На рис. 1 (см. вторую сторону обложки) представлена модель только одного процесса модели "Страхова-

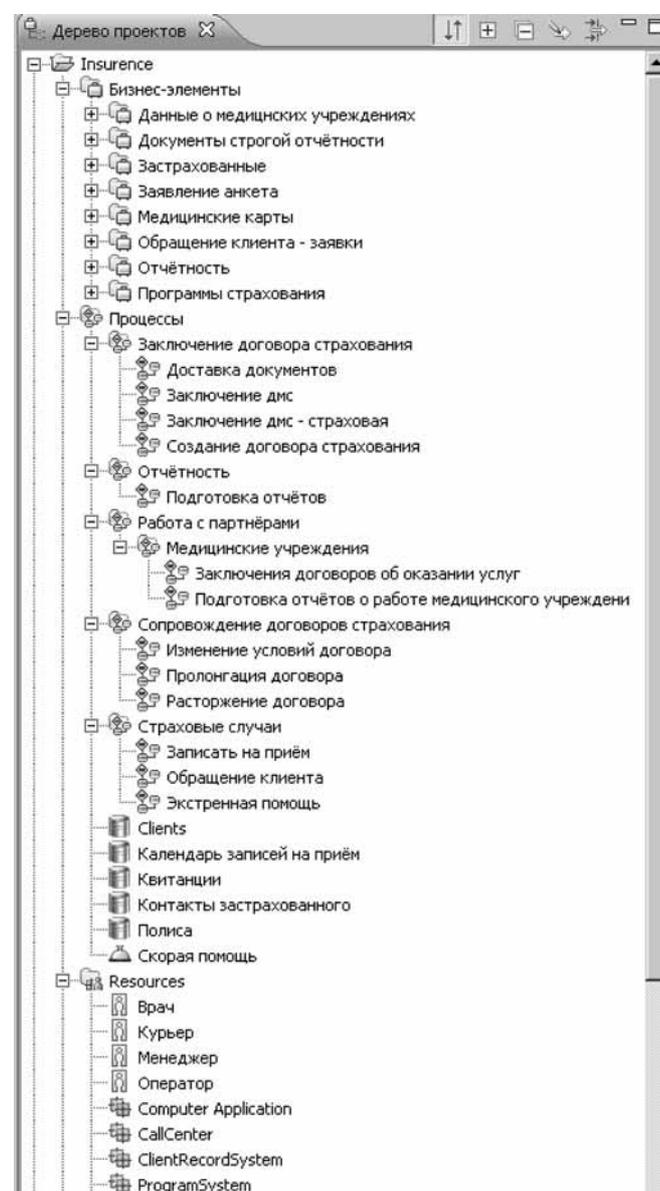


Рис. 2. Проект модели страхования в браузере проектов

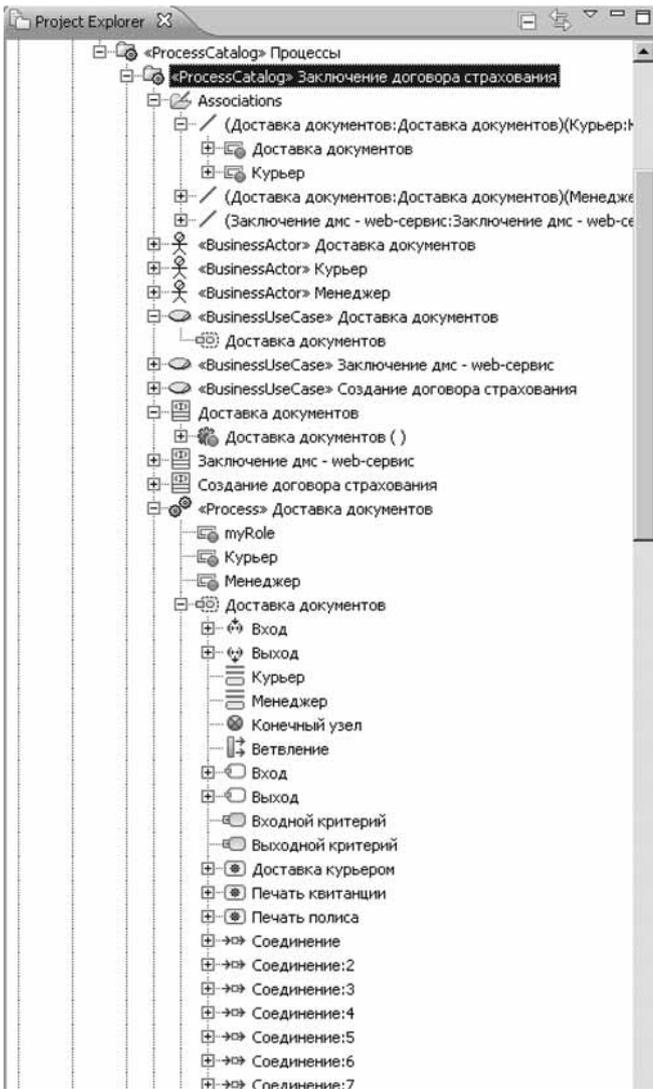


Рис. 4. Элементы папки "ProcessCatalog"Процессы проекта модели страхования в браузере проектов



Рис. 5. Элементы папки "serviceModel"Insurance проекта модели страхования в браузере проектов

ние" (Insurance), а именно процесса "Заклучение дмс" (заклучение договора медицинского страхования).

Сервисная архитектура информационной системы создается в среде продукта RSA. Этими вопросами ве-  
 дает архитектор. В отсутствии бизнес-моделей архи-  
 тектору постоянно необходимо сопоставлять бизнес-  
 представления с элементами компонентной модели.  
 Чтобы упростить или облегчить такое аналитическое  
 сопоставление, архитекторы обращаются к средствам  
 моделирования системы на более высоких уровнях  
 абстракции, чтобы определить рациональность при-  
 думываемых объектов (классов), методов, свойств  
 объектов объектно-ориентированной архитектуры  
 системы, преломляя при этом ее в сервисную ориен-  
 тацию. Архитектор должен обладать особым умением  
 и богатым опытом проводить такой анализ и разра-  
 ботку [3].

На рис. 2 представлена нотация бизнес-модели в  
 браузере проектов продукта WebSphere Business  
 Modeler.

При наличии у архитектора готовой бизнес-моде-  
 ли (см. рис. 1, вторая сторона обложки) существенно  
 понижается сложность решения проблемы сервис-  
 ориентированной декомпозиции системы в условиях  
 автоматической трансформации этой модели в сер-  
 висную модель. Это преимущество обеспечивается  
 путем импортирования бизнес-модели, полученной в  
 среде WebSphere Business Modeler, в среду разработки  
 продукта RSA. Действия по импортированию приве-  
 дены на рис. 3, см. вторую сторону обложки.

Прервемся немного от основной мысли о транс-  
 формациях и обратим внимание на появившиеся ана-  
 литические возможности для архитектора.



Рис. 6. Тематическая диаграмма в среде RSA

Рис. 3 хорошо демонстрирует эти возможности RSA в контекстном меню, представляя широкий диапазон аспектов анализа при визуализации разных видов модели, включая Topic Diagram.

Сервис-ориентированная декомпозиция средств IBM Rational [4] осуществляется в два этапа.

1. Импорт бизнес-модели, полученной в среде WebSphere Business Modeler, в среду разработки продукта RSA и получение BusinessUseCase модели, т. е. объектной модели. Само импортирование не создает сразу сервисную модель. В результате импортирования создается бизнес-модель в соответствии со стандартом UML2, так называемая BusinessUseCaseModel или BusinessAnalysisModels.

2. Трансформация BusinessUseCase модели в BusinessAnalysisServices модель, т. е. в сервис-ориентированную модель.

На рис. 4 демонстрируются элементы папки "ProcessCatalog"Процессы, как результат трансформации бизнес-модели в объектную модель деятельности медицинского страхования в браузере проектов. Сравнивая эти элементы с элементами модели на рис. 2, можно заметить, что все они присутствуют в обеих моделях, только в разных соответствующих друг другу графических нотациях. Более глубокое соответствие трудно проследить из-за ограниченной визуализации рисунков в статье.

Наряду с возможностями трансформации бизнес-модели при ее импортировании RSA позволяет осуществлять следующую трансформацию: из модели BusinessUseCaseModel в модель BusinessAnalysisServicesModels, представленную на рис. 5.

Данной трансформацией формируется сервисная модель: папка "serviceModel" и проект WebServiceProject, которая подготовлена, чтобы провести следующую трансформацию для получения Java-каркасов. Получение Java-каркасов в данной статье не рассматривается.

Если сравнить элементы моделей, представленные на рис. 2, рис. 4 и рис. 5, то можно заметить, что уже на уровне абстракции бизнес-модели закладывает-

ся сервисная ориентация архитектуры системы. В браузере проектов (Project Explorer), раскрыв все элементы папки "ProcessCatalog"Процессы, можно наблюдать результаты трансформации бизнес-модели в объектную модель, а раскрыв все элементы папки "serviceModel"Insurence, можно наблюдать результаты трансформации объектной модели BusinessUseCaseModel в сервисную модель, каждая со своей модифицированной графической нотацией элементов полученной модели, соответствующей элементам бизнес-модели в среде WebSphere Business Modeler.



Рис. 7. Элементы папки Insurence в среде RSA

Последняя трансформация — трансформация объектной модели в сервисную модель — является многосторонней.

Во-первых, осуществляется генерация графической нотации сервисной модели в браузере проектов (см. рис. 5).

Во-вторых, осуществляется визуализация UML-модели (TopicDiagram), которая представлена на рис. 6.

В-третьих, еще более важным является то, что осуществляется генерация файлов, отвечающих спецификациям WSDL и XSD. С ними можно проводить дальнейшее совершенствование, перейдя в режим (перспективу) "Web" среды разработки RSA [5]. В данной статье уделим внимание только файлам WSDL (рис. 7).

Как видно на рис. 7, по всем элементам бизнес-модели, которые относятся к сервисно-ориентирован-

```
<?xml version="1.0" encoding="UTF-8"?><wsdl:definitions name="Заключениедмсwebсервис"
targetNamespace="http://Процессы/Заключениедоговорастрахования/Заключениедмсwebсервис/"
xmlns:tns="http://Процессы/Заключениедоговорастрахования/Заключениедмсwebсервис/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:types>
    <xsd:schema
targetNamespace="http://Процессы/Заключениедоговорастрахования/Заключениедмсwebсервис/"
xmlns:xsd_1="http://Бизнесэлементы/Заявлениеанкета/"
xmlns:xsd_2="http://Бизнесэлементы/Документыстрогойотчётности/">
      <xsd:import namespace="http://Бизнесэлементы/Заявлениеанкета/" schemaLocation="../../xsd-
includes/http.Бизнесэлементы.Заявлениеанкета.xsd"/>
      <xsd:import namespace="http://Бизнесэлементы/Документыстрогойотчётности/"
schemaLocation="../../Бизнесэлементы/Документыстрогойотчётности/Квитанция.xsd"/>
      <xsd:element name="Заключениедмсwebсервис">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="Вход" type="xsd_1:Заявлениеанкета"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="ЗаключениедмсwebсервисResponse">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="Выход" type="xsd_2:Квитанция"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:schema>
  </wsdl:types>
  <wsdl:message name="ЗаключениедмсwebсервисRequest">
    <wsdl:part element="tns:Заключениедмсwebсервис" name="parameters"/>
  </wsdl:message>
  <wsdl:message name="ЗаключениедмсwebсервисResponse">
    <wsdl:part element="tns:ЗаключениедмсwebсервисResponse" name="parameters"/>
  </wsdl:message>
  <wsdl:portType name="Заключениедмсwebсервис">
    <wsdl:operation name="Заключениедмсwebсервис">
      <wsdl:input message="tns:ЗаключениедмсwebсервисRequest"/>
      <wsdl:output message="tns:ЗаключениедмсwebсервисResponse"/>
    </wsdl:operation>
  </wsdl:portType>
</wsdl:definitions>
```

Рис. 8. Текст WSDL-файла, сгенерированного в среде RSA на основе трансформации BusinessUseCaseModel в модель BusinessAnalysisServicesModels

ной архитектуре информационной системы, сформированы WSDL-файлы, т. е. интерфейсы сервисов.

Если провести аналогию с генерацией каркасов кода объектно-ориентированных классов на основе UML-диаграмм в Rational Rose, то WSDL-файлы тоже являются продуктом кодогенерации. А если учитывать, что RSA позволяет осуществлять генерацию каркасов кода функциональности сервисов на основе WSDL-файлов, то в данном случае можно наблюдать возможность разработки сервис-ориентированной информационной системы средствами генерации сервисов в среде RSA на основе бизнес-моделей. Посмотрим более пристально на текст получаемых в результате генерации WSDL-файлов (рис. 8).

Конечно, сразу бросается в глаза, что файл не полностью отвечает спецификации WSDL. Например, нет раздела с названием сервиса. Кроме этого, файл WSDL не должен быть русифицированным, так как русскоязычные элементы файла WSDL не дадут возможности осуществлять генерацию Java-кодов. Можно сказать, что так же, как и в случае генерации каркасов кода функциональности на основе UML-диаграмм, в случае WSDL-файлов мы имеем дело с каркасами интерфейсов. Эти интерфейсы требуют доработки. Возможно, совершенствуя бизнес-модель, можно добиться более совершенного WSDL-файла.

Несмотря на это, главная цель рассмотренных трансформаций — решить проблему декомпозиции

информационной системы, соответствующей бизнес-процессам. Эта цель достигается, что отвечает основной концепции SOA — слияние бизнес-логики с логикой информационной технологии, реализующей эту бизнес-логику [6].

#### Список литературы

1. **Design SOA services with Rational Software Architect.** URL: [http://www.ibm.com/developerworks/views/rational/libraryview.jsp?search\\_by=Design+SOA+services+with+Rational+Software+Architect,+Part](http://www.ibm.com/developerworks/views/rational/libraryview.jsp?search_by=Design+SOA+services+with+Rational+Software+Architect,+Part)
2. **Arsanjani A.** Service-oriented modeling and architecture: How to identify, specify, and realize services for your SOA (Сервис-ориентированное моделирование и архитектура: как выполнить идентификацию, создание спецификации и реализацию сервисов для SOA). URL: <http://www.ibm.com/developerworks/ru/library/ws-soa-design1/>
3. **Amsden J.** Моделирование SOA. Часть 1. Идентификация сервиса. URL: [http://www.ibm.com/developerworks/ru/library/1002\\_amsden/](http://www.ibm.com/developerworks/ru/library/1002_amsden/)
4. **Amsden J.** Моделирование SOA. Часть 2. Спецификация сервиса. URL: [http://www.ibm.com/developerworks/ru/library/1009\\_amsden/](http://www.ibm.com/developerworks/ru/library/1009_amsden/)
5. **Amsden J.** Моделирование SOA. Часть 3. Реализация сервиса. URL: [http://www.ibm.com/developerworks/ru/library/1016\\_amsden/](http://www.ibm.com/developerworks/ru/library/1016_amsden/)
6. **Биберштейн Н., Боуз С., Джонс К., Фиаммант., Ша Р.** Компас в мире сервис-ориентированной архитектуры (SOA): ценность для бизнеса, планирование и план развития предприятия. Пер. с англ. М.: КУДИЦ-ПРЕСС, 2007. 256 с.

## ИНФОРМАЦИЯ



# 2012 Software Engineering CEE-SEC(R) Conference in Russia

1-2 ноября

Москва, Digital October



Александр  
Петренко  
ВМК МГУ  
им. М.В. Ломоносова



Билл  
Кертис  
Cast Software



Александр  
Галицкий  
Almaz Capital  
Partners

и еще более 60 спикеров

При ранней регистрации (до 1 сентября) стоимость участия - 12 000 рублей  
Один день участия - 8 000 рублей. Студентам и аспирантам скидка 50%  
Действуют групповые и партнерские скидки  
Подробнее: [www.secr.ru](http://www.secr.ru)

**С. В. Соловьев**, канд. техн. наук, доц., зам. нач. упр.,  
**И. В. Затока**, канд. техн. наук, вед. науч. сотр.,  
**О. П. Лавлинская**, науч. сотр,  
ГНИИИ ПТЗИ ФСТЭК России, г. Воронеж  
e-mail: O-Lavl@yandex.ru

## Особенности проектирования информационных систем обеспечения деятельности по технической защите информации

*Рассматривается возможность использования экспертных методов при проектировании информационных систем обеспечения деятельности по технической защите информации на примере информационных систем в области защиты персональных данных.*

**Ключевые слова:** персональные данные, информационная система персональных данных, экспертная система

Одной из определяющих тенденций развития современного общества является его повсеместная информатизация. Она представляет собой организационный социально-экономический и научно-технический процесс создания оптимальных условий для удовлетворения информационных потребностей и реализации прав граждан, органов государственной власти, органов местного самоуправления, организаций и общественных объединений на основе формирования и использования информационных ресурсов [1]. Государственная политика в сфере формирования информационных ресурсов и информатизации в настоящее время направлена на создание условий для эффективного и качественного информационного обеспечения решения стратегических и оперативных задач социального и экономического развития Российской Федерации.

Одновременно с активным внедрением информационных технологий в деятельность органов власти и организаций повышается уровень угроз безопасности информации, обрабатываемой в их информационных системах. В соответствии с доктриной [2] одним из основных направлений повышения уровня защищенности объектов информационно-технологической инфраструктуры является реализация комплексного подхода к решению задач обеспечения их информационной безопасности.

Деятельность по защите информации имеет сложную структуру и включает организационные, правовые, технические, технологические, информационные и другие составляющие. Она осуществляется в рамках многофункциональных организационно-технических систем технической защиты информации (ТЗИ), формируемых на государственном, ведомственном и региональном уровнях, а также на уровне отдельных предприятий и организаций. В Российской Федерации создаются и совершенствуются системы ТЗИ, призванные обеспечить защиту объектов информационно-технологической инфраструктуры.

Одним из условий развития и совершенствования систем ТЗИ является внедрение в их деятельность информационных технологий, использование которых обеспечивает автоматизацию процесса подготовки решений. В настоящее время широкое распространение получили используемые в различных областях деятельности информационные системы персональных данных (ИСПДн), которые содержат информацию о физических или юридических лицах. Утечка такой информации, а также ее потеря или несанкционированное изменение могут привести к невосполнимому ущербу, а порой и к полной остановке деятельности отдельных организаций. При этом могут затрагиваться интересы конкретного человека, что ведет к нарушению его конституционных прав на неприкосновен-

ность частной жизни, личную и семейную тайну. Понимая важность и ценность информации о человеке, а также необходимость соблюдения прав своих граждан, государство требует от организаций и физических лиц обеспечить надежную защиту персональных данных (ПДн).

Для приведения деятельности в области защиты прав субъектов ПДн в соответствие с нормами европейского законодательства был принят Федеральный закон № 152-ФЗ от 27.07.2006 "О персональных данных", который вступил в силу в январе 2007 г. В данном законе определены общие требования, которые позднее были конкретизированы в подзаконных актах Правительства Российской Федерации и Министерства связи, нормативно-методических документах ФСТЭК России, ФСБ России и Роскомнадзора. Каждый из этих актов и документов посвящен отдельным областям и тематикам законодательства в области защиты ПДн.

В Федеральном законе о персональных данных [3] определены понятия персональных данных, операторов, обрабатывающих эти данные, а также понятия, связанные с обработкой и распространением ПДн. Согласно этому документу операторами ПДн являются практически все организации, которые ведут свою деятельность на территории Российской Федерации, поскольку они, как минимум, осуществляют сбор, систематизацию и хранение сведений о своих сотрудниках, клиентах и партнерах. На операторов ПДн возлагаются определенные обязанности, важнейшей из которых является обеспечение безопасности ПДн. Этот факт означает, что оператор ПДн обязан принять все необходимые меры для обеспечения конфиденциальности (а в некоторых случаях доступности и целостности) сведений о субъектах ПДн. Решение задачи защиты ПДн требует учета большого числа факторов и знания документов, на основе которых выбираются и реализуются методы и способы защиты персональных данных.

В общем случае порядок защиты персональных данных заключается в следующем:

- проводится классификация ИСПДн согласно документу [4]. По результатам классификации ИСПДн относится к типовой или к специальной, для типовой системы определяется ее класс;
- для типовых ИСПДн в зависимости от класса информационной системы, определенного в соответствии с документом [4], методы и способы защиты согласно классу определяются на основе положения [5];
- для специальных информационных систем методы и способы защиты формируются на основе определяемых оператором (уполномоченным лицом) угроз безопасности ПДн (модели угроз); модель угроз разрабатывается на основе методических документов, утвержденных в соответствии с п. 2 постановления Правительства Российской Федерации от 17 ноября 2007 г. № 781.

Выбранные и реализованные методы и способы защиты информации в информационной системе

должны обеспечивать нейтрализацию потенциальных угроз безопасности ПДн при их обработке в ИСПДн.

Так, в ИСПДн должно быть обеспечено:

- предотвращение несанкционированного доступа к ПДн и (или) передачи их лицам, не имеющим права доступа к такой информации;
- своевременное обнаружение фактов несанкционированного доступа к ПДн;
- недопущение воздействия на технические средства автоматизированной обработки ПДн, в результате которого может быть нарушено их функционирование;
- возможность незамедлительного восстановления ПДн, модифицированных или уничтоженных вследствие несанкционированного доступа к ним;
- постоянный контроль над обеспечением уровня защищенности ПДн.

Одним из первых шагов, направленных на определение методов и способов защиты информации, необходимых для обеспечения безопасности ПДн в процессе создания информационных систем или в ходе их эксплуатации (для уже введенных в эксплуатацию или модернизируемых систем), является классификация ИСПДн. Проведение классификации ИСПДн включает в себя следующие этапы:

- сбор и анализ исходных данных по информационной системе;
  - присвоение информационной системе соответствующего класса и его документальное оформление.
- При проведении классификации ИСПДн учитываются:
- категория ПДн, обрабатываемых (планируемых к обработке) в ИСПДн;
  - объем обрабатываемых ПДн;
  - заданные оператором характеристики безопасности ПДн, обрабатываемых в информационных системах;
  - структура информационных систем;
  - наличие подключений к сетям связи общего пользования и (или) к сетям международного информационного обмена;
  - режим обработки ПДн;
  - режим ограничения прав доступа пользователей информационных систем;
  - местонахождение технических средств информационных систем.

Классификация ИСПДн является одним из важных элементов работы, так как от класса информационной системы зависят требования по ее защите. Алгоритм отнесения ИСПДн к тому или иному классу представлен на рис. 1.

Согласно данному алгоритму ИСПДн классифицируются как типовые или как специальные. Если ИСПДн классифицируется как типовая, то в этом случае ее защита осуществляется в строгом соответствии с требованиями ФСТЭК России, предъявляемыми для различных классов ИСПДн [5]. Если ИСПДн классифицируется как специальная, то для

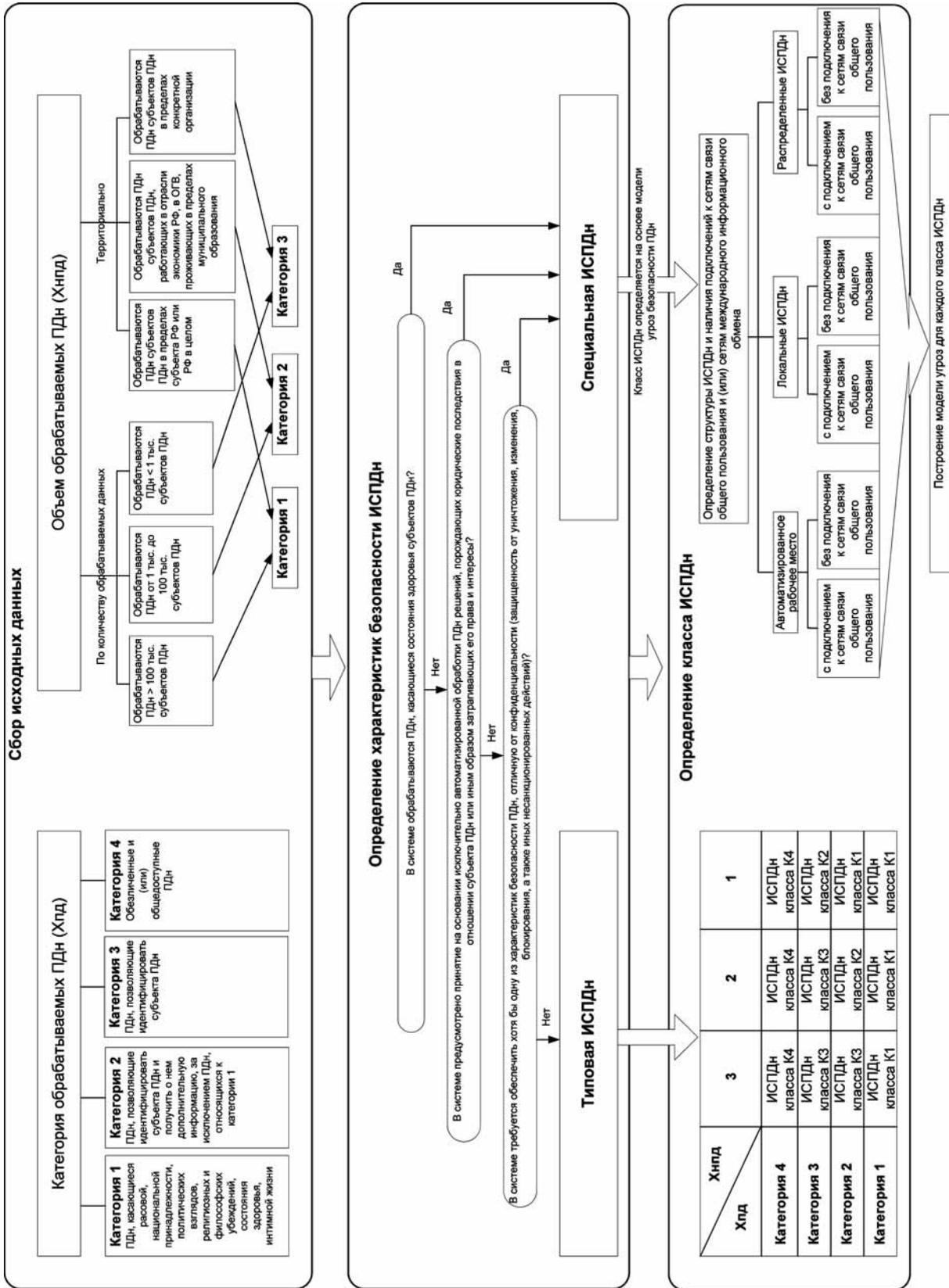


Рис. 1. Алгоритм проведения классификации ИСПДн

нее разрабатывается актуальная модель угроз, на основании которой определяются меры по обеспечению защиты ПДн.

Чтобы разработать систему защиты ИСПДн, операторам информационной системы необходимо знать требования, которые определены в руководящих документах [5], а также уметь применить эти требования на практике.

Анализ представленного на рис. 1 алгоритма показал, что уже на начальном этапе формирования способов и методов защиты ПДн, при проведении классификации ИСПДн операторы ИСПДн сталкиваются со сложностью алгоритма и необходимостью учета достаточно большого числа факторов, определяющих решения по защите информации. Кроме того, они должны владеть существующими методами и способами защиты информации, а также знать об имеющихся сертифицированных средствах защиты информации.

Практика показала, что реализовать требования Федерального закона "О персональных данных" самостоятельно организациям (специалистам, эксплуатирующим ИСПДн, операторам ИСПДн, не являющимся специалистами в области защиты информации) достаточно сложно, так как в случае неверной интерпретации требований руководящих документов существуют следующие риски:

- увеличение сроков и стоимости реализации проекта по защите ПДн в случае чрезмерной защиты ПДн в результате завышения класса ИСПДн;
- риск отрицательных результатов проверок, выявления нарушений и, как следствие, наложение соответствующих санкций в случае необоснованного снижения класса ИСПДн.

Принимая во внимание, что постановка задачи определения класса ИСПДн, а также обоснования методов и способов защиты информации, обрабатываемой в ИСПДн, в силу объективных причин может обладать определенной неполнотой исходных данных, неоднозначностью трактовки отдельных положений, для ее решения предлагается использовать экспертные методы, такие как анкетирование, интервьюирование, дискуссия. Выбор того или иного метода определяется полнотой, достоверностью и соответствием исходной информации целевым установкам задачи, временем и ресурсозатратами, которые выделяются на ее решение.

Реализация перечисленных выше методов указывает на необходимость использования экспертных систем, отличительной особенностью которых является их ориентация на решение недостаточно формализуемых задач.

Экспертные системы обладают следующими особенностями [6]:

- алгоритм решений неизвестен заранее, а строится самой экспертной системой с помощью символических рассуждений, базирующихся на эвристических приемах;

- система способна объяснить, как было получено решение;
- система обладает способностью приобретения новых знаний от пользователя-эксперта, не владеющего в необходимом объеме методами программирования;
- обеспечение "дружественного", как правило, естественно-языкового интерфейса с пользователем.

Создание экспертной системы предполагает использование группы экспертов для формирования ее основных компонентов, а именно базы знаний и механизма вывода. Для этого целесообразно использовать приведенные выше экспертные методы. С этой целью осуществляется опрос персонала ИСПДн, специалистов в области защиты информации, должностных лиц, в результате которого, как правило, происходит выявление особенностей и условий функционирования ИСПДн и возможностей защиты ПДн при их обработке в информационных системах. Для проведения опроса составляются специальные опросные листы, вопросы для которых формируются в соответствии с существующими нормативными и нормативно-методическими документами в области защиты персональных данных.

Далее представлен примерный перечень вопросов, которые могут использоваться для определения класса экспертной системы.

1. Какие персональные данные предполагается обрабатывать в ИСПДн?
  2. К какой категории можно отнести обрабатываемые ПДн?
  3. Предполагаемый объем обрабатываемых ПДн?
  4. Предполагается ли обработка данных, касающихся состояния здоровья субъектов ПДн?
  5. Предполагается ли разграничение прав доступа пользователей при их работе с ИСПДн?
  6. Имеется ли подключение к сетям связи общего пользования?
  7. Необходимо ли обеспечивать защищенность ПДн от уничтожения, изменения, блокирования?
  8. Является ли ИСПДн распределенной?
  9. Является ли ИСПДн многопользовательской?
  10. Какие способы защиты информации уже используются в ИСПДн?
- и т. д.

При защите ПДн на экспертную систему возлагаются следующие задачи:

- классификация ИСПДн;
- выбор методов и способов защиты ПДн при их обработке в типовой ИСПДн;
- формирование модели угроз безопасности ПДн для специальной ИСПДн;
- выбор способов и методов защиты ПДн при их обработке в специальной информационной системе.

Для выбора адекватных методов и способов защиты персональных данных наиболее важными компонентами экспертной системы являются база знаний и механизм логического вывода, обеспечивающие для операторов ИСПДн подготовку решений по защите

таких данных. Формирование этих компонентов экспертной системы должно осуществляться на основе нормативных и правовых документов, регламентирующих деятельность по защите ПДн при их обработке в ИСПДн.

Одним из важных вопросов при построении экспертной системы, предназначенной для выбора способов защиты информации, обрабатываемой в ИСПДн, является определение способа представления знаний. Его важность определяется тем обстоятельством, что форма представления знаний оказывает существенное влияние на характеристики и свойства системы. Представление знаний должно быть понятным как экспертам, так и операторам ИСПДн, пользователям системы. К основным моделям представления таких знаний относятся [7]:

- логическая модель;
- модель, основанная на использовании правил (продукционная модель);
- модель, основанная на использовании фреймов;
- модель семантической сети.

В основе построения экспертной системы для рассматриваемой предметной области целесообразно использовать продукционную модель представления знаний, что определяется следующими факторами [7]:

- большая часть знаний в рассматриваемой предметной области может быть записана в виде продукций;
- системы продукций являются модульными, поэтому в большинстве случаев удаление или добавление продукций не приводит к изменениям в остальных продукциях;
- при необходимости системы продукций могут реализовать любые алгоритмы.

В целом существующие в настоящее время подходы к проектированию сложных систем можно разделить на два больших класса. К первому из них относится структурный (системный) подход или анализ, основанный на идее алгоритмической декомпозиции, где каждый модуль системы выполняет один из важнейших этапов общего процесса. Второй класс представляет объектный подход, связанный с декомпозицией и выделением не процессов, а объектов. Каждый объект при этом рассматривается как экземпляр определенного класса. Перспективы этого подхода обусловлены тем обстоятельством, что он унифицирует процесс создания программных средств. В нем более полно реализуется технология структурного программирования, облегчается процесс создания сложных иерархических систем, появляется удобная возможность создания пользовательских библиотек объектов в различных областях применения.

Для формирования базы знаний рассматриваемой продукционной экспертной системы возможно использование обобщенного объектно-структурного подхода, который подразумевает интегрированное использование преимуществ обоих подходов, что позволяет реализовать более гибкую стратегию организации ветвления программы (передачи управления), которое должно управляться самими данными. При

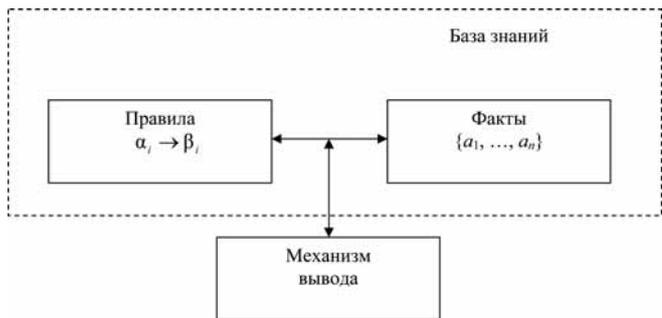


Рис. 2. Продукционная экспертная система

этом достигается логическая цельность и ясность программы, что важно как для понимания ее работы во время создания самого программного продукта, так и для его последующей модификации в процессе эксплуатации.

Далее в качестве примера приведена возможность использования продукционной модели представления знаний при классификации ИСПДн. Продукционная экспертная система классификации ИСПДн, как и любая продукционная система, состоит из трех основных компонентов: набора правил, используемого в качестве базы знаний (базы правил); рабочей памяти (или памяти для кратковременного хранения), в которой хранятся предпосылки, касающиеся конкретных задач рассматриваемой предметной области (факты), и результаты выводов, полученных на их основании; механизма логического вывода, использующего правила в соответствии с содержимым рабочей памяти (рис. 2).

Правила-продукции  $\alpha_i \rightarrow \beta_i$  интерпретируются с помощью конструкции:

ЕСЛИ  $\alpha_i$ , то  $\beta_i$ .

Например, для определения категории обрабатываемых ПДн ( $X_{\text{пд}}$ ) действуют следующие правила.

**Правило 1.** ЕСЛИ в ИСПДн обрабатываются данные, касающиеся расовой принадлежности,

ТО  $X_{\text{пд}} = 1$ .

**Правило 2.** ЕСЛИ в ИСПДн обрабатываются данные, касающиеся национальной принадлежности,

ТО  $X_{\text{пд}} = 1$ .

**Правило 3.** ЕСЛИ в ИСПДн обрабатываются данные, касающиеся политических взглядов,

ТО  $X_{\text{пд}} = 1$ .

**Правило 4.** ЕСЛИ в ИСПДн обрабатываются данные, касающиеся религиозных и философских убеждений,

ТО  $X_{\text{пд}} = 1$ .

**Правило 5.** ЕСЛИ в ИСПДн обрабатываются данные, касающиеся состояния здоровья,

ТО  $X_{\text{пд}} = 1$ .

**Правило 6.** ЕСЛИ в ИСПДн обрабатываются данные, позволяющие идентифицировать субъекта ПДн с получением о нем дополнительной информации,

И информация о субъекте ПДн не относится к  $X_{\text{пд}} = 1$ ,  
ТО  $X_{\text{пд}} = 2$ .

**Правило 7.** ЕСЛИ в ИСПДн обрабатываются данные, позволяющие идентифицировать субъекта ПДн без получения о нем дополнительной информации,

ТО  $X_{\text{пд}} = 3$ .

**Правило 8.** ЕСЛИ в ИСПДн обрабатываются обезличенные данные,

ТО  $X_{\text{пд}} = 4$ .

**Правило 9.** ЕСЛИ в ИСПДн обрабатываются общедоступные данные,

ТО  $X_{\text{пд}} = 4$ .

По такому же принципу формируются правила для определения объема обрабатываемых ПДн ( $X_{\text{нпд}}$ ), характеристик безопасности ИСПДн, структуры ИСПДн и т. д. Факты, которые записываются в рабочую память, представляют собой образцы в виде наборов символов (например, "назначение ИСПДн — обработка данных о состоянии здоровья" и т. п.). После того как в рабочую память записываются факты, рассматривается возможность применения правил.

Механизм вывода в продукционной экспертной системе будет строиться следующим образом.

Пусть для рассматриваемой экспертной системы имеется базовое множество фактов  $\{a_1, \dots, a_n\}$ , формируемое при разработке экспертной системы. Задача экспертной системы состоит в установлении связей между фактами для конкретной ситуации, интересующей пользователя (оператора ИСПДн).

Различают два подмножества  $A_0$  и  $A_1$  исходного множества  $A$ :

$$A = A_0 \cup A_1, A_0 \cap A_1 = \emptyset,$$

где  $A_1$  — множество констатированных (помеченных) фактов;  $A_0$  — множество непомеченных фактов.

На начальном этапе работы экспертной системы множество  $A_1$  содержит некоторое количество фактов (исходной информации), например

$$A_1 = \{a_1, a_2, a_3\}.$$

Далее происходит последовательное пополнение множества  $A_1$  за счет элементов множества  $A_0$ . Интерпретатор правил сопоставляет левые части продукций  $\alpha_i \rightarrow \beta_i$  с имеющимися в множестве  $A_1$  фактами и выполняет то правило, левая часть которого  $\alpha_i$  согласуется с фактами из  $A_1$  (оказывается истинной). В результате множество  $A_1$  пополняется за счет фактов, констатируемых в правой части продукции  $\beta_i$ .

Пусть имеется некоторая ИСПДн, для которой требуется провести классификацию. Для данной ИСПДн определено исходное множество фактов  $A_1$ , состоящее из элементов

$$A_1 = \{a_1, a_2\}.$$

Интерпретатором правил в базе знаний находятся продукции  $P$ :

$$P_1: a_1 \rightarrow a_3;$$

$$P_2: a_2 \rightarrow a_4.$$

Тогда, согласно работе [8], продукции выполняются и новое множество  $A_1$  имеет состав

$$A_1 = \{a_1, a_2, a_3, a_4\}.$$

Далее процесс повторяется, т. е. проверяется согласованность с  $A_1$  оставшихся продукций.

Таким образом, осуществляется последовательное определение правил-продукций, позволяющее по исходным данным получить интересующий пользователя факт (в данном примере — установить класс ИСПДн).

Процесс сопоставления левых частей продукций  $\alpha_i$  с множеством констатированных фактов  $A_1$  порождает цепочку вывода или цепочку рассуждений [8]. Графически эту цепочку можно изобразить, как показано на рис. 3.

Данная цепочка вывода соответствует последовательному применению продукций

$$a_1 \rightarrow a_3;$$

$$a_2 \rightarrow a_4;$$

$$a_3 \wedge a_4 \rightarrow a_5$$

к исходному множеству  $A_1 = \{a_1, a_2\}$ . Новый факт  $a_5$  и есть результат, интересующий пользователя.

Таким образом, продукции  $\alpha_i \rightarrow \beta_i$  можно трактовать как способ описания шагов общения между пользователем и экспертной системой. Результат выполнения некоторого правила, например, может означать вопрос к пользователю, ответ на который приведет к пополнению множества  $A_1$ . Использование продукционной модели возможно также при выборе методов и способов защиты типовой ИСПДн.

В связи с большим объемом знаний и необходимостью решения более сложных задач при формиро-

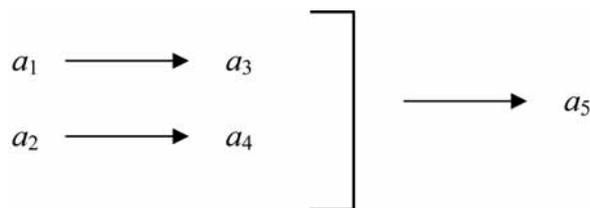


Рис. 3. Цепочка вывода

вании защиты специальной ИСПДн возможно использование других моделей представления знаний, таких как фреймвые модели, семантические сети, логические модели либо их комбинации. База знаний, кроме характеристики информационной системы и обрабатываемых в ней ПДн, должна содержать информацию о существующих в настоящее время методах и способах защиты, сертифицированных средствах защиты информации, а также документах, которыми операторы ИСПДн должны руководствоваться при формировании способов защиты ПДн при их обработке в ИСПДн. Интегрируя такую информацию, экспертная система может выполнять и функции информационной или информационно-справочной системы.

По своему назначению и функциональным возможностям инструментальные программы, применяемые при проектировании экспертных систем, можно разделить на четыре категории:

- оболочки экспертных систем;
- языки программирования высокого уровня;
- среда программирования;
- дополнительные модули.

Проведенный авторами анализ современных средств разработки экспертных систем показал целесообразность использования для разработки экспертной системы, рассматриваемой в данной работе, многофункциональной среды программирования CLIPS (*C Language Integrated Production System* — язык С, интегрированный с продукционными системами), представляющей собой современное средство, предназначенное для создания экспертных систем, и содержащее большое число готовых инструментальных средств (встроенный механизм управления базой знаний, механизм логического вывода, менеджеры различных объектов CLIPS и подобных им) и конструкций (упорядоченные факты, шаблоны, правила, функции, родовые функции, классы, модули, ограничения, встроенный язык COOL и др.). Существующая в настоящее время версия CLIPS может эксплуатироваться на платформах UNIX, DOS, Windows и Macintosh. Она является хорошо документированным общедоступным программным продуктом. В связи с тем что среда программирования CLIPS обладает

расширяемостью, мощностью и низкой стоимостью, она получила широкое распространение в государственных организациях, индустрии и учебных заведениях.

Таким образом, приведенные выше алгоритмы и правила вывода составляют основу проектирования экспертной системы в области защиты ПДн. Отличия в проектировании экспертных систем, используемых для защиты ИСПДн в разных областях деятельности, будут заключаться в объеме обрабатываемых ПДн, их категории, а также в наборе правил и механизме логического вывода для конкретной ИСПДн.

В связи с тем обстоятельством, что в настоящее время ИСПДн нашли широкое применение в деятельности органов государственной и исполнительной власти различных уровней, а операторы ИСПДн далеко не всегда имеют квалифицированных специалистов в области защиты информации, задача создания экспертных систем, обеспечивающих выбор эффективных методов и способов защиты ИСПДн, является очень важной. В этой связи представленные выше предложения являются практически значимыми и могут использоваться при разработке данных экспертных систем.

#### Список литературы

1. **Об информации**, информатизации и защите информации. Федеральный закон от 20 февраля 1995 г. № 24-ФЗ.
2. **Доктрина** информационной безопасности Российской Федерации. Утверждена Указом Президента Российской Федерации от 9 сентября 2000 г.
3. **О персональных данных**. Федеральный закон от 27 июля 2006 г. № 152-ФЗ.
4. **Порядок** проведения классификации информационных систем персональных данных. Утвержден приказом ФСТЭК России, ФСБ России, Мининформсвязи России от 13 февраля 2008 г. № 55/86/20.
5. **Положение** о методах и способах защиты информации в информационных системах персональных данных. Утверждено приказом ФСТЭК России от 5 февраля 2010 г. № 58.
6. **Искусственный интеллект**: Кн. 1. Системы общения и экспертные системы / Под ред. Э. В. Попова. М.: Радио и связь, 1990. 464 с.
7. **Искусственный интеллект**: Кн. 2. Модели и методы / Под ред. Д. А. Поспелова. М.: Радио и связь, 1990. 304 с.
8. **Черноруцкий И. Г.** Методы принятия решений. СПб.: БХВ-Петербург, 2005. 416 с.

**И. Ю. Жуков**<sup>1</sup>, д-р техн. наук, проф.,  
**Д. М. Михайлов**<sup>2</sup>, канд. техн. наук., доц., e-mail: mr.mdmitry@gmail.com,  
**А. В. Кукушкин**<sup>2</sup>, студент,  
**А. В. Стариковский**<sup>2</sup>, науч. сотр.,  
**Е. В. Иванова**<sup>3</sup>, врач-эндоскопист,  
**Е. Д. Федоров**<sup>3</sup>, д-р мед. наук, клинический руководитель отд. эндоскопической хирургии,  
**А. М. Толстая**<sup>2</sup>, студент,  
<sup>1</sup>ФГПУ "ЦНИИ ЭИСУ", г. Москва,  
<sup>2</sup>НИЯУ МИФИ, г. Москва,  
<sup>3</sup>Городская клиническая больница № 31, г. Москва

## Программные методы распознавания участков желудочно-кишечного тракта, полипов и кровотечений в беспроводной капсульной эндоскопии

*Описаны модули, повышающие эффективность распознавания патологий и морфологических изменений в желудочно-кишечном тракте (ЖКТ), а также его секций по изображениям, полученным с беспроводной эндоскопической капсулы. В основе метода распознавания кровотечений и секций ЖКТ лежит пиксельный анализ изображения. Обнаружение полипов происходит посредством анализа выпуклости поверхности. Строятся 3D-модели, измеряется кривизна поверхности. В результате анализа алгоритм определяет точки в исходном изображении, которые указывают на наличие полипа.*

**Ключевые слова:** капсульная эндоскопия, кровотечение, полип, секции, яркость, насыщенность, пиксель, анализ, кривизна

Беспроводная капсульная эндоскопия — это современный метод неинвазивной диагностики желудочно-кишечного тракта (ЖКТ) человека, позволяющий визуализировать области, недоступные для традиционных методов гастро- и колоноскопии.

Автоматический анализ гастроэнтерологического тракта и его болезней представляет большой интерес благодаря его потенциалу в распознавании патологий и морфологических изменений. Рентгеновское обследование и другие неинвазивные методы исследования могут определять лишь грубую патологию.

По сравнению с традиционной эндоскопией, у беспроводной есть важное преимущество: капсула может достичь двенадцатиперстной и тонкой кишок, не доставляя при этом дискомфорта пациенту. Более

того, это единственный нехирургический метод осмотра тонкой кишки [1].

Процедура проводится следующим образом: пациент проглатывает маленькую капсулу габаритами 11 × 26 мм, которая благодаря перистальтическим сокращениям медленно проходит сквозь ЖКТ, захватывая изображения со скоростью два кадра в секунду. Изображения имеют размер 256 на 256 пикселей. Капсула оснащена автономным освещением, что позволяет получать изображения ЖКТ достаточно высокого качества. Записанные изображения передаются на переносной жесткий диск, находящийся у пациента на поясе, с использованием специальных систем и протоколов передачи изображений с эндоскопической капсулы. Исследование длится около восьми ча-

сов, в течение которых пациент может заниматься обычными повседневными делами [1–4].

За восемь часов запоминающее устройство накапливает более 57 тыс. изображений, которые по окончании исследования передаются на стационарный компьютер для дальнейшего анализа. Обычно на просмотр изображений врач-гастроэнтеролог тратит около двух часов. Чтобы уменьшить это время, в программу, через которую врач просматривает эти изображения, встраиваются модули автоматического распознавания патологий [2, 4].

Данная статья посвящена описанию трех модулей, определяющих участки, наличие полипов и наличие кровотечений в ЖКТ человека.

### Распознавание участков желудочно-кишечного тракта

Метод распознавания участков ЖКТ основан на способе получения информации о местоположении капсулы с помощью цветового анализа изображения, полученного на этом участке. Вначале выявляются опорные изображения, а затем определяется участок ЖКТ.

Как уже было сказано выше, результатом гастроэнтерологического исследования являются несколько тысяч снимков ЖКТ пациента. В данном случае необходимо реализовать метод, который позволяет удалять лишние изображения, слабо отличающиеся от соседних, а затем выделять опорные участки, которые и будут анализироваться врачом.

Разработанный метод заключается в следующем. Все полученные изображения преобразуются в цветовую схему HSV (*Hue, Saturation, Value* — тон, насыщенность, яркость), и выполняется построение нормированных гистограмм цветовой насыщенности (*Saturation*) для каждого из них.

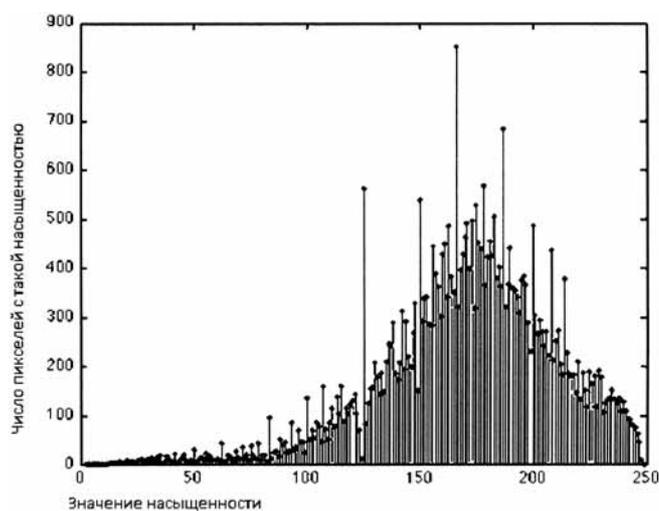
По некоторым причинам на гистограмме в значениях насыщенности 0 и 255 столбцы будут значительно выше, чем в остальных. Проведенные эксперименты показали, что это верно для всех изображений. Поэтому при нормировании значения в точках 0 и 255 просто отбрасываются, так как не являются информативными. Таким образом, для всех изображений получают гистограммы, которые впоследствии сравниваются.

При создании метода применялась идея, которую использовал Пирсон при описании распределения хи-квадрат [5]. Суть ее состоит в том, что повторяющиеся подряд факторы (тон, насыщенность, значение), встречающиеся реже, имеют больший вес (т. е. числовое значение), чем те, которые встречаются чаще. Анализируя полученные веса при сравнении гистограмм соседних изображений, можно выделить коэффициент схожести изображений друг на друга (отношение числовых значений соседних изображений). Для этого числовое значение первого изображения делится на числовое значение второго, значение второго — на значение третьего и так далее последовательно.

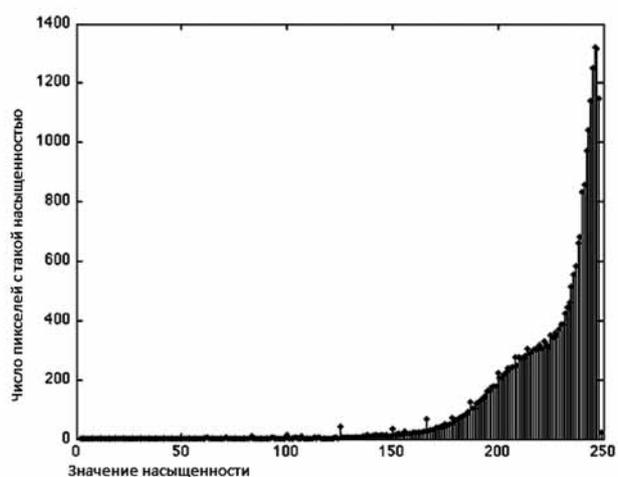
Считается, что чем коэффициент меньше, тем изображения более похожи. Например, два изображения, коэффициент схожести которых равен нулю, идентичны.

Определив пороговый коэффициент  $k_e$ , можно выявить, какие изображения являются опорными, а какие можно не рассматривать. Метод предполагает, что первое изображение помечается как опорное, а каждое последующее сравнивается с выбранным. Если коэффициент схожести  $k$  больше порогового  $k_e$ , то выявляется новое опорное изображение. Последующие изображения сравниваются уже с ним.

Получив опорные изображения, можно перейти к целевому этапу. Здесь снова проводится анализ гистограмм цветовой насыщенности: они сравниваются с эталонными гистограммами отделов ЖКТ. На рис. 1 приведены примеры эталонных гистограмм цветовой насыщенности желудка (рис. 1, а) и кишечника (рис. 1, б) из тестируемой выборки. Тестируемая вы-



а)



б)

Рис. 1. Эталонные гистограммы цветовой насыщенности желудка (а) и кишечника (б)

борка представляет собой реальные изображения, полученные в результате капсульного исследования ЖКТ.

На практике при определении участков ЖКТ можно увидеть некий "шум": например, в группе изображений желудка алгоритм может найти несколько изображений из кишечника. Таким образом, существует вероятность, что изображение из кишечника может быть распознано как изображение из желудка. Этого можно избежать, если программно учитывать, из какого участка были получены соседние снимки, а также тот факт, что исследование начинается с желудка, и капсула перемещается только в одном направлении. С данными ограничениями можно довольно точно определить зоны ЖКТ, представленные на снимках.

Таким образом, обработка данных, полученных с помощью беспроводной капсульной эндоскопии, может быть значительно упрощена за счет автоматизации процесса отбора снимков, пригодных для исследования.

### Распознавание полипов в желудочно-кишечном тракте

Метод распознавания полипов в ЖКТ включает в себя описание математической модели модуля с дальнейшим сглаживанием возникающего "шума".

Предположим, что имеется цветное изображение, полученное эндоскопической капсулой, которое содержит полип. Оно переводится в черно-белую цветовую схему. В результате этой операции получается изображение, приближенное к тому, что приведено на рис. 2.

Математическая модель подразумевает, что полип является выпуклостью по отношению к окружающему ландшафту и представляет собой идеальную сферу с однородной поверхностью. На изображении центр

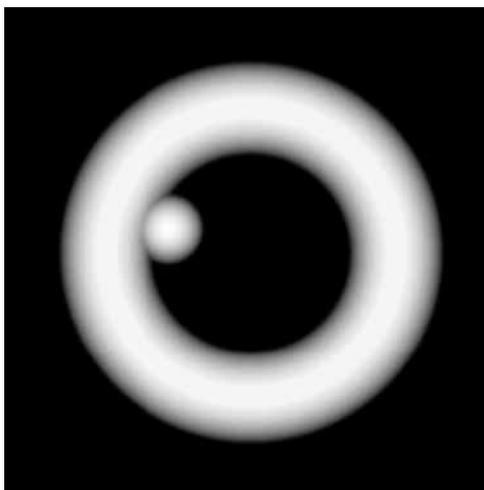


Рис. 2. Предполагаемая приближенная модель полипа

круга имеет яркость значительно большую, чем по краям.

Изображение представляет собой таблицу пикселей. Каждый пиксель — это число (чем ярче пиксель, тем больше его значение). После определения численного значения каждого пикселя изображение анализируется с точки зрения его трехмерного представления. При этом значение яркости в определенной точке есть ее высота в третьем измерении. Таким образом, можно сделать переход от двухмерного изображения к трехмерному представлению исследуемого участка ЖКТ.

Далее определяется местоположение полипа, если таковой вообще имеется. Для этого измеряется кривизна поверхности в каждой точке полученного трехмерного пространства. По формулам (1) подсчитываются кривизна Гаусса ( $K$ ) и средняя кривизна ( $M$ ) в каждой точке модели [6, 7].

$$K = \frac{F_{xx}F_{yy} - F_{xy}^2}{(1 + F_x^2 + F_y^2)^2}, \quad (1)$$

$$M = \frac{(1 + F_{xx})^2 F_{yy} - 2F_x F_y F_{xy} + (1 + F_{yy})^2 F_{xx}}{2\sqrt{(1 + F_x^2 + F_y^2)^3}}(x, y),$$

где  $F_x^2$ ;  $F_{xx}$ ;  $F_y^2$ ;  $F_{yy}$ ;  $F_{xy}$  — производные в точке по соответствующим осям.

Производные берутся с помощью оператора Собеля. Для вычисления производной  $F_{xy}$  последовательно применяется оператор Собеля для оси  $y$  и для оси  $x$  [6]. Результирующие производные показаны на рис. 3, см. третью сторону обложки.

После получения производных вычисляются кривизна Гаусса и средняя кривизна для каждой точки изображения по формулам (1) (рис. 4, см. третью сторону обложки) [7]. Кривизна Гаусса — естественная мера кривизны. Ее значение зависит только от того, как расстояния измеряются на поверхности.

Таким образом, формируются два двумерных массива  $G$  (Gaussian) и  $M$  (Mean) размерностью  $m \times n$  ( $m$  — число пикселей в исходном изображении по горизонтали;  $n$  — по вертикали). Элементами массива  $G_{(i,j)}$  являются значения гауссовской кривизны в точках  $(i, j)$  (аналогично для массива  $M$ ).

Значение массива  $G$  тем больше, чем сильнее изогнута поверхность, оно всегда положительно. Значение массива  $M$  положительно, если поверхность выпуклая, и отрицательно, если поверхность вогнута [7]. Так как полип — это выпуклость, рассматриваются только значения, отвечающие параметру "выпуклые".

Из полученных массивов  $G$  и  $M$  формируется массив  $P$  (polyp curvature), элементы которого определяются по следующему закону:

$$P_{(i,j)} = -G_{(i,j)} \min(0; M_{(i,j)}). \quad (2)$$

Элемент  $(-\min(0, M_{(i, j)}))$  превращает в ноль все положительные значения  $M$  и делает все отрицательные  $M$  положительными, т. е. значение  $P$  — полипная кривизна — получается большой и положительной для больших выпуклостей и нулевой для любых вогнутых участков.

Если какой-то элемент массива  $P$  превышает заранее заданную границу  $\alpha$ , то это означает, что в исходном изображении в точке, соответствующей этому элементу, находится полип (рис. 5, см. третью сторону обложки).

Однако в реальном случае при переводе цветовой схемы изображения в черно-белую становится заметно так называемое зерно или шум. Это очень часто чередующиеся точки со сравнительно большим и маленьким значением яркости. При переводе в трехмерную модель эти точки превратятся в ступень с очень большой разницей по высоте, что, в свою очередь, скажется на вычисленной кривизне. Это может привести к обнаружению полипа там, где его на самом деле нет. Чтобы это предотвратить, используется сглаживание (*Gaussian blur*). Оно применяется как к исходному изображению, так и к массивам производных, необходимых для вычисления кривизны [8].

Для сглаживания использовался фильтр Гаусса (*Gaussian Blur*). У фильтра Гаусса есть два параметра —  $r$  и  $\sigma$ ,  $\sigma$  был сделан линейнозависимым от  $r$ , поэтому дальше речь пойдет только об  $r$ . Физический смысл  $r$  такой: чем  $r$  больше, тем меньше шума и сигнала в изображении. Поэтому нужно выбрать такое значение  $r$ , чтобы сохранить детали изображения и при этом уменьшить шум. Параметр для сглаживания изображения обозначим как  $r_I$ , параметр для сглаживания производных —  $r_{dI}$ . Чтобы определить значения  $r_I$  и  $r_{dI}$ , была посчитана ошибка алгоритма на обучающей выборке при различных значениях  $r_I$  и  $r_{dI}$  и были выбраны значения, дающие минимальную ошибку. Использовались значения:  $r_I = 3$ ,  $r_{dI} = 27$ ,  $\alpha = 0,9$ .

С учетом описанных ограничений был реализован программный модуль, обнаруживающий выпуклости на снимках с беспроводной эндоскопической капсулы и показывающий их местоположение на снимке.

### Распознавание кровотечений в желудочно-кишечном тракте

Исследование медицинских изображений часто требует применения ряда техник анализа, учитывающих различные особенности изображений. Нужно также принимать во внимание анатомическую и патологическую важность просматриваемых участков.

Предлагаемый метод распознавания кровоточащих областей разделен на два основных шага. Первый шаг отделяет фрагменты видео, содержащие признаки кровотечения, от несодержащих. На втором шаге происходит дальнейший анализ изображений кровотечений, подтверждающий, действительно ли данные изображения содержат признаки активного кровотечения (рис. 6).

При просмотре гастроэнтерологических изображений наиболее характерный признак, определяющий

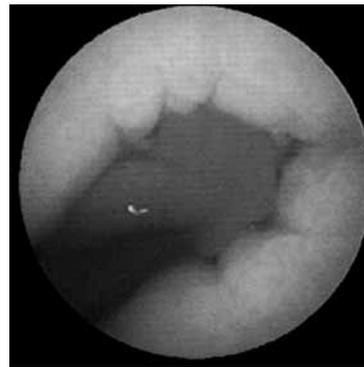


Рис. 6. Изображение, содержащее кровотечение

присутствие кровотечения, — это яркий красный цвет или присутствие темно-красных областей [9]. Важно отметить, что в большей части ЖКТ превалирует красный цвет. Но красные оттенки в некровоточащих участках имеют меньшую цветовую насыщенность, чем в кровоточащих. Автоматическое распознавание кровоточащих областей можно свести к распознаванию участков изображения с насыщенным красным цветом.

Цветовую насыщенность и оттенок можно легко узнать, преобразовав цветовую схему изображения в HSV. Через  $I_H(x, y)$  обозначим значение тона ( $H$ ) канала изображения  $I$  в точке  $(x, y)$ , аналогично для насыщенности ( $I_S$ ) и яркости ( $I_V$ ). Значение каждого параметра канала кодируется одним байтом, поэтому  $I_S, H, V \in [0, 255]$ . Как упоминалось ранее, в большинстве гастроэнтерологических изображений присутствуют цвета с красным оттенком. Цвета, превалирующие в кишечнике, варьируются от оранжевого до желто-зеленого из-за присутствия остатков фекалий, что делает распознавание кровотечений более трудным. Однако цветовая насыщенность почти не зависит от расположения капсулы.

Для получения первичной классификации (содержит изображение кровотечения или не содержит) используется блочный подход, который позволяет быстро проанализировать все видео [10]. Каждое изображение разделяется на блоки размером  $64 \times 64$  пикселя, как показано на рис. 7, и для каждого блока  $B_i$  подсчитывается среднее значение цветовой насыщенности.

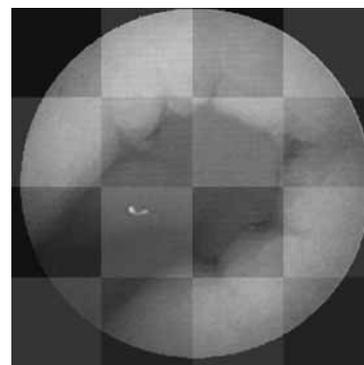


Рис. 7. Изображение, разбитое на блоки

Блоки, располагающиеся в углах изображения, содержат очень мало интересующих нас пикселей; их можно считать не содержащими кровотечение. Затем алгоритм находит второй по насыщенности блок из двенадцати оставшихся ( $B_2 - \min$ ). Блок с наименьшей насыщенностью не рассматривается, так как обычно он соответствует просвету в ЖКТ, через который продвигается капсула. Изображение распознается как содержащее кровотечение, если хотя бы один блок, содержащий кровь, удовлетворяет следующему условию:

$$\text{Кровотечение есть, если } \bar{I}_s(B_i) < \bar{I}_s(B_2 - \min) + m,$$

где  $\bar{I}_s(B)$  — среднее значение цветовой насыщенности в блоке  $B$ , а  $m$  — граница надежности. Во время тестирования она устанавливалась равной  $0,15 \times 255$ .

В случае несоответствия изображения данному условию можно считать, что кровотечения оно не содержит. На рис. 8 приведена визуализация цветовой насыщенности в каждом блоке исходного изображения. Чем блок светлее, тем цветовая насыщенность пикселей, содержащихся в нем, больше. Блок, соответствующий  $\bar{I}_s(B_2 - \min)$ , отмечен рамкой.

Однако некоторые изображения, не содержащие реальных признаков кровотечения, могут быть ошибочно классифицированы как содержащие кровотечение. Далее они будут перепроверены, а их статус пересмотрен.

На втором шаге рассматриваются только те эндоскопические изображения, которые были помечены как содержащие кровотечение. Проводится дальнейший анализ изображений, который подтверждает или опровергает решение, полученное на первом шаге.

Здесь проводится пиксельный анализ яркости и цветовой насыщенности, тогда как предыдущий шаг оперировал только значениями средней насыщенности в крупных блоках ( $64 \times 64 = 4096$  пикселей в каждом блоке).

Кровотечения в изображениях с беспроводной эндоскопической капсулы обычно соответствуют диапазону относительно небольших значений яркости. Первичный анализ показал, что кровотокающие зоны чаще всего содержат значения яркости пикселей между 30 и 80. В результате можно вычислить два уровня границ —  $T_l$  в следующем условии соответствует нижней границе,  $T_h$  — верхней.

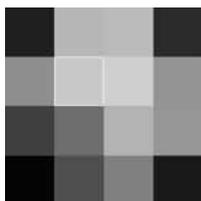


Рис. 8. Блоки с подсчитанными значениями средней цветовой насыщенности

Кровотечение есть:

$$T_h > I_V(x, y) > T_l.$$

Кровотечения нет: во всех остальных случаях.

Однако вследствие того что освещение различных участков может быть разным, выставление фиксированных границ в эндоскопических изображениях не всегда корректно. Таким образом, границы лучше задавать динамически. Параметр  $\bar{I}_V$ , соответствующий среднему пиксельному значению яркости для каждого изображения, используется для автоматической установки границ [9].

$$T_h = \bar{I}_V + 10,$$

$$T_l = \bar{I}_V - 30.$$

Цветовая насыщенность снова используется здесь, но уже для пиксельного анализа. Стоит заметить, что для хорошо освещенных изображений, в которых доминирует красный цвет, цветовая насыщенность может достигать очень больших значений. Это основная причина ошибок, связанных с пропуском кровотечения на изображении, реально его содержащем. Кровотокающие участки с высокой насыщенностью выявляются благодаря применению следующего условия:

Кровотечение есть:

$$0,95 \times 255 > I_s(x, y) > 0,7 \times 255. \quad (2)$$

Кровотечения нет: во всех остальных случаях.

Предложенная классификация включает комбинацию анализов по яркости и насыщенности. Окончательное решение зависит от того, как эти результаты соотносятся с различными условиями освещения. Оно может быть выражено с помощью нижеописанных условий.

1.  $\bar{I}_V < 60$ : изображение имеет недостаточную освещенность, следовательно, эффективным будет анализ только по критерию насыщенности.

2.  $\bar{I}_V \in [60, 75]$ : критерии яркости и насыщенности используются в комбинации. Классификация кровотокающих требует совпадения по обоим критериям.

3.  $\bar{I}_V > 75$ : критерии яркости и насыщенности используются в комбинации, но из-за того что изображение слишком яркое, границы неравенства (2) меняются:  $0,95 \times 255 > I_s(x, y) > 0,85 \times 255$ .

Результат работы второго шага алгоритма показан на рис. 9.

Для того чтобы сделать классификацию более точной, изображения разделяются по уровню интенсивности кровотока. Используется четырехуровневая классификация, в которой каждому уровню соответствует определенный цветовой код (см. таблицу). Для этих целей анализируется общее количество пикселей  $N_{S2}$ , отмеченных после шага 2 как кровотечение.  $N_{S2}$  сравнивается с  $N_{\min}$  параметром, используемым для разграничения изображений по уровням. По умолчанию значение  $N_{\min}$  установлено равным 450, если был

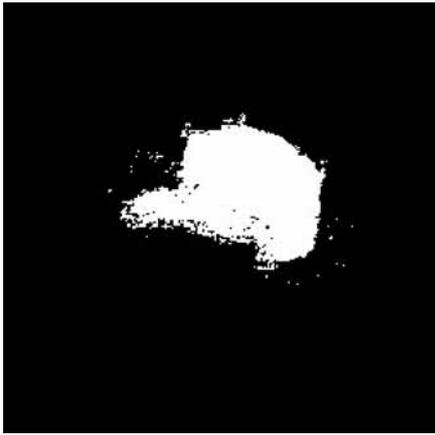


Рис. 9. Иллюстрация второго шага распознавания: белые пиксели соответствуют кровотечению

#### Классификация уровней кровотечения

Уровень	0	1	2	3
Цветовой код				

применен только критерий цветовой насыщенности. При применении обоих критериев, насыщенности и яркости, параметр выставляется равным 5000. Решение о принадлежности изображения, содержащего кровотечение, принимается по следующим правилам.

1. Уровень 0 (кровотечение отсутствует):

$$N_{S2} < N_{\min}$$

2. Уровень 1 (кровотечение малой интенсивности):

$$N_{\min} < N_{S2} \leq 1,8N_{\min}$$

3. Уровень 2 (кровотечение умеренной интенсивности):

$$1,8N_{\min} < N_{S2} \leq 2,6N_{\min}$$

4. Уровень 3 (кровотечение большой интенсивности):

$$N_{S2} > 2,6N_{\min}$$

Метод распознавания кровотечений в ЖКТ был экспериментально проверен на материалах тестируемой выборки. Были проанализированы 72 изображения, из которых 22 содержали кровотечение и 50 были без признаков кровотечения. В результате эксперимента была выявлена следующая неточность: на трех

изображениях из тестируемой выборки, содержащих кровотечение, алгоритм их не выявил; на 13 изображениях алгоритм выявил наличие кровотечения, хотя его там не было.

Эта неточность вызвана значительным числом используемых констант, несмотря на это планируется дальнейшее усовершенствование алгоритма и повышение его точности. Кроме того, на точность выполнения алгоритма также влияет качество камеры, принимающей изображение ЖКТ.

Таким образом, беспроводная капсульная эндоскопия позволяет не только целиком и полностью исследовать ЖКТ, но и автоматически определить его секции, а также выявлять кровотечения и такие тканевые образования как полипы. Для этого используются математические методы, преобразующие получаемые с капсулы изображения в программную модель. Данная технология также позволит значительно сократить время, необходимое врачам для анализа изображений, определения патологий и постановки диагноза.

#### Список литературы

1. Waterman M., Eliakim R. Capsule enteroscopy of the small intestine. *Abdom Imaging*. 2008, 405: Epub ahead of print.
2. Жуков И. Ю., Кукушкин А. В., Михайлов Д. М., Семенов С. В., Стариковский А. В., Толстая А. М. Определение участков желудочно-кишечного тракта в беспроводной капсульной эндоскопии // Научная Сессия НИЯУ МИФИ-2012. Аннотации Докладов. Т. 2. Проблемы Фундаментальной Науки. Стратегические Информационные Технологии. М.: НИЯУ МИФИ, 2012. С. 235.
3. Козлов Н. А., Шурыгин В. А., Жуков И. Ю., Федоров Е. Д., Иванова Е. В., Михайлов Д. М. Метод сжатия изображений для беспроводной капсульной эндоскопии // Спецтехника и связь. 2011. № 6. С. 34—37.
4. Иванова Е. В., Жуков Ю. И., Федоров Е. Д. Программное обеспечение для программно-аппаратного эндоскопического капсульного комплекса для проведения безболезненного исследования желудочно-кишечного тракта, включая области, недоступные с помощью методов традиционной эндоскопии. Свидетельство об официальной регистрации программы для ЭВМ № 2011610692. Зарегистрировано в реестре программ для ЭВМ 11.01.2011.
5. Бостанджиян Б. А. Распределение Пирсона, Джонсона, Вейбулла и обратное нормальное. Оценивание их параметров. Редакционно-издательский отдел ИПХФ РАН, 2009. 240 с.
6. Engel K., Hadwiger M., Kniss J. M., Rezk-Salama C., Weiskopf D. *Real-Time Volume Graphics*. A K Peters, 2006. 497 p.
7. Kühnel W. *Differential Geometry: Curves — Surfaces — Manifolds*. American Mathematical Society, 2006.
8. Nixon M. S., Aguado A. S. *Feature Extraction and Image Processing*. Academic Press, 2008. 88 p.
9. Lau P. Y., Correia P. L. Detection of bleeding patterns in WCE video using multiple features // Proc. of the 29th Annual International Conference of the IEEE EMBS: Cite Internationale. Lyon, France. August 23—26, 2007. P. 5601—5604.
10. Paragios N., Chen Y., Faugeras O. *The Handbook of Mathematical Models in Computer Vision*. Springer, 2005. 596 p.

**Е. Н. Черемисина**, д-р техн. наук, директор, Институт системного анализа и управления Международного университета природы, общества и человека "Дубна",

**М. А. Белов**, доц., Международный университет природы, общества и человека "Дубна", НИУ ВШЭ,

**О. Е. Антипов**, асп., Институт системного анализа и управления Международного университета природы, общества и человека "Дубна",

**А. В. Сорокин**, канд., техн. наук., доц., менеджер университетских проектов, IBM в России и СНГ, e-mail: alexander\_sorokin@ru.ibm.com

## **Инновационная практика компьютерного образования в университете "Дубна" с применением виртуальной компьютерной лаборатории на основе технологии облачных вычислений**

*Рассказано о передовом опыте подготовки высококвалифицированных специалистов с применением виртуальной компьютерной лаборатории на основе сервис-ориентированной архитектуры и технологий облачных вычислений. В статье рассмотрены ключевые компоненты виртуальной компьютерной лаборатории и вопросы их интеграции. Представлена информация об аппаратном и программном обеспечении ядра системы, а также учебно-методических аспектах ее успешного использования в учебном процессе.*

**Ключевые слова:** виртуальная компьютерная лаборатория, технологии облачных вычислений, методика изучения корпоративного программного обеспечения, подготовка высококвалифицированных ИТ-специалистов

В настоящее время в научных исследованиях и образовании, в производственной и других сферах деятельности человека определяющее значение имеют информационно-вычислительные системы. Постоянное развитие науки, техники и технологий влечет за собой появление новых средств вычислительной техники, автоматизированных систем. Вследствие этого на рынке труда появляются новые вакансии, что увеличивает потребность людей в эффективном образовании, включая высшее, дополнительное, профессиональное, переподготовку и повышение квалификации. Образовательные услуги в таком контексте становятся очень востребованными. Однако образование не всегда идет в ногу со временем, выстраивая модели обучения в традиционной форме и не обеспечивая, тем самым, получение необходимых и актуальных знаний, умений и навыков [1].

Система высшего профессионального образования в настоящее время требует новых форм и методов преподавания. Инновации в образовательной деятельности,

комплексная модернизация системы образования — это, пожалуй, наиболее важные вопросы, которым в последнее десятилетие уделено немало внимания. Рассматриваются стратегические программы реформирования образования, исследуются стимулы его развития в долгосрочной перспективе и возможности интеграции системы отечественного образования в международное образовательное пространство. Исследователи считают, что ключевым направлением модернизации образования станет использование новых информационных технологий, компьютеризация учебных заведений и инновационная деятельность профессорско-преподавательского состава вузов [2].

### **Что такое инновационное образование?**

Инновационное образование — это создание в вузе оптимальной и устойчивой учебно-организационной, научно-методической и нормативно-административной среды. Такая среда обеспечивает поддержку инноваци-

онных подходов к образовательному процессу, которые ориентированы на интеграцию научно-образовательного потенциала вуза и отраслевой академической науки, на установление партнерских отношений с работодателями [3]. Инновационное образование подразумевает не только использование возможностей международного сотрудничества, непосредственной заинтересованности, инициативности профессорско-преподавательского состава и развитие творчества, усиление самостоятельности у студентов, но и применение зарубежного опыта. Такой подход подразумевает интеграцию в учебный процесс самых современных методик, форм обучения, частичную или полную автоматизацию всего образовательного процесса путем внедрения современных информационных и инновационных технологий и систем.

Основной принцип инновационного образования заключается в использовании новых опережающих знаний и основанных на них учебно-методических технологий будущего. Инновационное образование ориентировано не столько на передачу знаний, которые имеют свойство становиться неактуальными, сколько на овладение базовыми компетенциями, умениями и навыками, которые затем по мере необходимости позволяют приобретать новые знания, дополнительные умения и навыки уже самостоятельно. В этом заключается его отличие от традиционного. Снижение конкурентоспособности традиционных образовательных институтов, а также недостаточная интеграция новых знаний и технологий в образование и другие сферы экономики, как следствие недостатка компетентных специалистов, свидетельствуют о необходимости создания учреждений высшего образования принципиально нового типа. Сегодня традиционное образование, как система получения знаний, отстает от реальных потребностей современной экономики, в том числе науки, техники и промышленности [3].

Система образования в инновационном вузе должна быть открыта современным и перспективным научным исследованиям, а технологическое оснащение учебного процесса должно соответствовать уровню передовой науки. В учебном плане такого вуза должны присутствовать мастер-классы, тренинги, проектные разработки, стажировки на производстве и в научно-исследовательских организациях. Причем выполняться это должно для студентов любой формы обучения — очной, заочной и дистанционной. Внедрение инновационной практики преподавания только для одной из форм обучения в системе высшего и дополнительного образования в современных условиях обеспечит необходимое качество обучения лишь для некоторого контингента обучаемых. Однако такой подход не позволит раскрыть потенциал и развить личные качества учащихся других форм обучения.

Очень важно организовать учебный процесс таким образом, чтобы учащиеся могли эффективно получать актуальные знания, умения и навыки согласно выбранной ими индивидуальной траектории, а также предоставить им возможность лучше сориентироваться в основных видах социальной активности. Такая организация должна способствовать формированию компетенций, которые необходимы для выполнения типовых видов деятельности каждым гражданином в

рамках высокотехнологичного информационного общества, помочь осознанию личных и социальных возможностей, их реализации в социуме, осмыслению своей социальной принадлежности, обогащая содержание и формы общения, развивая умение общаться, обеспечивая выбор форм сотрудничества, формируя новый взгляд на самооценку и адаптацию к существующим реалиям.

Наиболее успешными в плане обеспечения инновационного характера развития образовательной деятельности становятся такие вузы, в которых одновременно реализуется совокупность следующих учебно-методических компонентов:

- разработка студентами реальных проектов в различных научных и производственных секторах;
- проведение исследований фундаментального и прикладного характера;
- использование образовательных технологий, обеспечивающих студентам возможность выбора учебных курсов;
- возможности создания и поддержки индивидуальных траекторий обучения для студентов очной, заочной и дистанционной форм обучения;
- предоставление студентам удаленного доступа посредством локальной сети и сети Интернет к программно-аппаратным ресурсам и программно-технологическим платформам университета для решения всевозможных учебных, научно-исследовательских и вычислительных задач;
- внедрение систем оценки качества работы, контроля усвояемости и полноты знаний.

Формирование инфраструктуры инновационного образования может происходить в несколько этапов, поэтапно, как в виде специальных подразделений вуза, так и в форме самостоятельного юридического лица.

### **Инновационные технологии компьютерного образования в университете "Дубна"**

Отличительной чертой современного этапа развития образования является большая составляющая в процессе обучения информационных технологий — ИТ-образование. Оно отличается непрерывным и быстрым ростом требований к квалификации специалистов, поскольку в последние десятилетия темпы обновления образовательного ресурса постоянно увеличиваются и сокращается период обновления знаний. Соответственно, постоянно растут и изменяются требования к системе ИТ-образования.

В настоящее время образовательная модель университета "Дубна" использует системный, проблемно-ориентированный и информационный подходы. Данные подходы определяют компетенции выпускников, владеющих современными методами системного анализа, навыками применения соответствующих теоретических разработок, методических материалов, технологий и компьютерных средств при решении конкретных задач. Эти выпускники должны не только обладать достаточной квалификацией для использования корпоративного программного обеспечения, но также иметь обширные знания, навыки и опыт разра-

ботки, установки и поддержки информационных систем любого масштаба.

Начиная уже с первого курса обучение студентов университета "Дубна" нацелено на приобретение навыков постановки предметных задач, на их структуризацию и формализацию, на системный анализ предметной области, поиск решений различных задач и оценку эффективности принятых решений. Подходы, используемые в образовательном процессе университета, позволяют студентам осваивать методы и средства решения конкретных задач и вести научно-исследовательскую деятельность, которая реализуется в ходе выполнения курсовых работ по различным дисциплинам [3].

Особенности IT-образования в университете "Дубна" подразумевают развитие конструктивного, аналитического и системного мышлений, что помогает самостоятельно формулировать и решать нестандартные задачи с позиции требований современных процессов, происходящих в социуме, ориентирует на понимание места и роли моделей и информационных систем, объектов и процессов в природе и обществе.

Принципы образовательной модели университета "Дубна" включают:

- фундаментальную университетскую подготовку;
- профессиональную подготовку по вопросам взаимодействия природы, общества и человека;
- информационную поддержку учебного процесса;
- предоставление студентам любой формы обучения программно-аппаратных и информационных ресурсов университета, аккумулирующих знания, в том числе в режиме удаленного доступа к ним, за счет внедрения в образовательный процесс инновационных решений.

Технологическая и информационная глобализация диктуют международные стандарты и требования к квалификации IT-специалистов и, соответственно, к национальной системе подготовки кадров [3]. Высокая скорость обновления технологий превращает IT-образование в постоянный непрерывный процесс. Такой процесс сопровождает современного IT-специалиста на протяжении всей профессиональной деятельности, оказывая непосредственное влияние на его карьерный рост и значимость для отрасли, в которой он работает.

Важным моментом современного IT-образования является обучение студентов — будущих бакалавров и магистров техники и технологии — обширному кругу информационных и корпоративных систем как проприетарных, так и свободно распространяемых. Это позволяет сформировать более широкое пространство знаний и навыков будущего специалиста, заинтересовать его многообразием информационных систем различного масштаба и назначения, обеспечить большей свободой их выбора. Сегодня не обязательно следовать за коммерческим программным обеспечением, привязывая учебный процесс к конкретным программным продуктам, приобретать лицензии на каждую рабочую станцию, нет необходимости в модернизации компьютерного парка университета и в покупке мощных рабочих станций его сотрудниками и студентами. Эти возможности могут быть реализованы с использованием инновационных программно-тех-

нологических разработок, современных программно-аппаратных платформ и высоконадежных центров обработки данных, которые внедряются в учебный процесс. Свобода действий вне зависимости от места и времени обучения, свобода выбора — это реалии сегодняшних инновационных технологий в образовании.

Для обеспечения качественного, мобильного и гибкого образования в университете "Дубна" разработаны и внедрены такие проекты по инновационной подготовке IT-специалистов, как виртуальная компьютерная лаборатория на основе технологии облачных вычислений и виртуальный компьютерный класс.

### **Предпосылки создания программно-технологической платформы виртуальной компьютерной лаборатории в университете "Дубна"**

В 2007 г. между университетом "Дубна" и корпорацией IBM был подписан Меморандум о взаимопонимании и создании Академического Центра Компетенции IBM в области проектирования виртуальных организаций. Основная задача центра заключается в проведении и поддержке научных некоммерческих исследований в области проектирования виртуальных организаций как новой формы ведения бизнеса и обучения в XXI веке. Кроме того, в рамках Академического Центра Компетенции IBM решаются следующие задачи:

- освоение и внедрение в практику учебного процесса университета открытых программных продуктов и информационных технологий корпорации IBM, предоставляемых университету в рамках программы академической инициативы, для профильных специальностей и направлений подготовки выпускников;
- подготовка предложений для стажировки преподавателей, аспирантов и студентов университета в рамках программ, проводимых корпорацией IBM;
- подбор кандидатур студентов, магистров и аспирантов университета для грантов корпорации IBM и последующей работы в компании;
- помощь студентам и аспирантам университета в выполнении научно-практических работ и исследований по тематике корпорации IBM.

Сотрудничество с Академическим Центром Компетенции предполагает, что студент знакомится с широким спектром корпоративных приложений фирмы IBM и изучает современные информационные технологии. Преподавателю такое сотрудничество позволяет бесплатно получать программное обеспечение IBM и активно использовать его в учебном процессе с учебными пособиями и методическими материалами по изучению и освоению современных информационных технологий. В рамках такого сотрудничества университету "Дубна" была предоставлена аппаратная платформа для научных исследований, размещения ресурсоемких информационных систем и работы с ними — центр обработки данных blade-архитектуры IBM BladeCenter S с тремя установленными на шасси blade-серверами.

Другим фактором, способствующим появлению программно-технологической платформы виртуальной

компьютерной лаборатории в университете "Дубна", стало следующее обстоятельство. Структура традиционных систем, используемых для электронного обучения как в России, так и за рубежом, как правило, включает большие, достаточно сложные в работе и установке комплексы, требующие для сопровождения выделенные серверы или мощные рабочие станции. По этой причине пользователям (и не только студентам вузов), заинтересованным в получении новых знаний и опыта в различных IT-областях, зачастую оказывается сложно установить необходимое программное обеспечение на своем персональном компьютере и не только из-за недостатка базовых знаний. На настоящее время существуют следующие трудности, связанные с самостоятельным развертыванием такого рода систем.

♦ Стоимость программного обеспечения оказывается высокой для отдельно взятого пользователя, и в большинстве случаев лицензия необходима лишь на время учебного процесса.

♦ Аппаратные требования корпоративных программных продуктов, как правило, выходят за рамки типовых домашних, офисных и переносных компьютеров.

♦ Возникновение конфликтов с предустановленными домашними развлекательными и познавательными программными продуктами и, как следствие, существенное снижение производительности.

Следует отметить, что сейчас большинство студентов вынуждены учиться без отрыва от работы. По этой причине они самостоятельно формируют индивидуальные траектории обучения, которые подразумевают удобное для них место и время обучения, например, в командировке. Благодаря успешному внедрению беспроводных телекоммуникационных интернет-технологий (3G, WiMax/LTE, iBurst) во многих крупных городах России и за рубежом обучение в любом месте и в любое время уже стало реальностью [1].

Другая распространенная трудность, возникающая в процессе обучения, заключается в том, что на домашних или переносных компьютерах учащихся, как правило, отсутствуют достаточные аппаратные и программные ресурсы, необходимые для развертывания клиент-серверных приложений, которые предназначены для освоения спецкурсов в рамках самостоятельной работы. Как следствие, это обстоятельство затрудняет выполнение внеаудиторных работ, курсовых и дипломных проектов. Особенно остро такие вопросы стоят перед студентами из других городов, проживающих в общежитиях, а также для студентов, обучающихся дистанционно, когда отсутствуют очные семинары и нет возможности прослушать некоторые спецкурсы. В этой ситуации выполнение лабораторных работ и практических заданий на домашних компьютерах фактически невозможно.

Чтобы разрешить отмеченные выше вопросы, необходимо было разработать систему, архитектура и технологические возможности которой позволяли бы пользователям-студентам обучаться в режиме удаленного доступа к учебным материалам с использованием браузера. Авторами в рамках университетской инициативы были успешно разработаны концепция виртуальной компьютерной лаборатории на основе технологий облачных вы-

числений, ее архитектура, информационное, техническое, технологическое и методическое обеспечение. Отметим, что за рубежом такая задача решается, например, в ходе эксперимента по развитию Виртуальной облачной лаборатории (*Virtual Cloud Laboratory*) [8], созданной в штате Северная Каролина (США).

### **Виртуальная компьютерная лаборатория на основе технологии облачных вычислений и ее роль в учебном процессе университета "Дубна"**

*Виртуальная компьютерная лаборатория на основе технологии облачных вычислений* представляет собой комплекс программно-аппаратных средств, основанный на технологиях виртуализации, позволяющих гибко, по запросу пользователя предоставлять ему вычислительные ресурсы университета для создания виртуальных серверов, выполнения научно-исследовательских работ, ресурсоемких вычислительных расчетов и заданий, связанных с освоением сложных корпоративных и иных информационных систем.

Программно-аппаратная платформа виртуальной лаборатории на основе технологии облачных вычислений, которая внедрена в учебный процесс университета "Дубна", позволяет предоставлять студентам удаленный доступ к необходимым вычислительным ресурсам для выполнения заданий, связанных с освоением сложных корпоративных систем и обеспечением доступа к базовым приложениям, находящимся в университете, независимо от местоположения студента. Виртуальная компьютерная лаборатория может быть эффективно использована студентами для решения различных учебных, научно-исследовательских и вычислительных задач. В их числе могут быть: решение курсовых и лабораторных работ; создание и сопровождение баз данных, веб-серверов и различных клиент-серверных приложений; ведение дипломных проектов; поддержка системы дистанционного обучения и т. п.

Разработанный и внедренный образец виртуальной компьютерной лаборатории обладает следующими особенностями:

- интернет-ориентированный доступ к ресурсам лаборатории, централизованная система управления всеми компонентами на основе интеграционного портала;
- поддержка учебного процесса с возможностью его оперативного изменения;
- высокая вычислительная и пропускная способности компонентов аппаратной платформы благодаря применению серверов blade-архитектуры и специализированной системы хранения данных;
- достаточная компактность системы и относительная неприхотливость к условиям функционирования;
- резервирование данных и всех жизненно важных компонентов системы;
- масштабирование и сервисное обслуживание в "горячем режиме";
- открытое платформообразующее программное обеспечение;
- балансировка нагрузки.

Основные особенности, которые диктуют высокие требования к аппаратному обеспечению виртуальной компьютерной лаборатории и используемым сетевым ресурсам, — это и большое число одновременно работающих в виртуальной среде клиентов, и необходимость наличия свободных системных ресурсов, которые могут быть задействованы в горячем режиме, по требованию. Резервирование данных и всех компонентов программно-аппаратной платформы позволяет поддерживать бесперебойную работу всей системы и проводить быструю замену неисправного оборудования. Резервирование питания и отработанные алгоритмы корректного завершения работы помогают предотвратить сбои, вызванные ошибками файловой системы в случае некорректного завершения работы функционирующего гипервизора и всех работающих виртуальных машин.

Применение открытого программного обеспечения позволяет по мере необходимости модифицировать его, совершенствовать виртуальную лабораторию, интегрировать ее с другими системами, которые применяются в учебном процессе.

Гибкость в управлении ресурсами и их распределение по запросу пользователя (работа в "облаке") позволяют автоматически управлять нагрузкой всех виртуальных машин и быстро выполнять наиболее ресурсоемкие приложения или вычисления [5].

Интернет-ориентированный подход предоставляет возможность не только дистанционно пользоваться ресурсами лаборатории, но и в режиме удаленного доступа управлять всеми аппаратными и программными компонентами лаборатории, выполнять настройку и проводить необходимые обновления. Специальный портал позволяет объединять программные компоненты управления в единую, централизованную систему. Основными компонентами управления в ее составе являются: консоль управления BladeCenter S; консоль управления дисковой подсистемой; подсистема управления распределенной сетевой файловой системой; подсистема управления сквозной аутентификацией и авторизацией; подсистема управления образами программного обеспечения; подсистема мониторинга; подсистема безопасности; система проведения мастер-классов и интернет-лекций; интерактивное электронное учебно-методическое интерактивное пособие; система совместной работы с документами и коммуникациями.

Использование серверной платформы blade-архитектуры позволяет устанавливать несколько гипервизоров ESX и ориентировать каждый blade-сервер на решение конкретных задач. Например, один из blade-серверов будет ориентирован на работу с виртуальным компьютерным классом на базе Citrix XenApp™, дидактическими и интерактивными учебными материалами и видеоуроками. Другой blade-сервер будет отвечать за научно-исследовательские работы, совмещая одновременно множество различных платформ (Windows, Unix, Linux, FreeBSD, MacOS) и их приложений на созданных виртуальных машинах. В виртуальной компьютерной лаборатории нет каких-либо ограничений на установку той или иной программной платформы или продукта. Компонентный подход позволяет адаптировать работу виртуальной компьютерной лаборатории

под особенности учебного процесса, оптимизируя тем самым соотношение цена—качество.

Созданное к настоящему времени программное ядро виртуальной компьютерной лаборатории выступает в роли оболочки для всего учебного процесса. Оно представляет собой универсальный инновационный инструмент, который, используя возможности современной сети Интернет, способен успешно выполнять все основные задачи, возникающие в учебном процессе, а именно:

- проводить оценку базовых знаний учащихся для определения уровня сложности заданий, формируя, по сути, индивидуальную траекторию самостоятельной практической работы в виртуальной компьютерной лаборатории;
- знакомить учащихся с теоретическим материалом, необходимым для выполнения практических заданий, проводить мастер-классы, выполнять ряд задач технической поддержки на основе интернет-ориентированных аудиовизуальных технологий (вебинаров);
- размещать и использовать на информационном интернет-ориентированном портале лаборатории интерактивные инструкции, методические пособия и мультимедийный контент для выполнения самостоятельных, лабораторных работ и иных заданий;
- изучать корпоративное программное обеспечение в различных операционных системах (Windows, Unix/Linux, FreeBSD, MacOS, DOS);
- моделировать работу распределенных систем, кластеров (*failover cluster*), систем балансировки нагрузки (NLB), Grid (на примере ORACLE RAC);
- выполнять научно-исследовательские работы и различные ресурсоемкие вычисления в рамках курсовых и дипломных работ, а также проводить самостоятельные исследования;
- контролировать и управлять работоспособностью и загрузкой как отдельных виртуальных машин, так и лабораторией в целом, включая все задействованные аппаратные ресурсы и компоненты;
- контролировать уровень усвоения знаний, обеспечивая непрерывное и эффективное управление процессом обучения.

В таблице приведены сравнительные показатели традиционной системы ИТ-образования и инновационной системы с интегрированной в учебный процесс виртуальной компьютерной лабораторией.

В разработанной виртуальной компьютерной лаборатории на основе технологии облачных вычислений, которая успешно внедрена в учебный процесс университета "Дубна", используется свободно распространяемое программное обеспечение. В первую очередь это альтернативные Microsoft Windows операционные системы, а именно Unix, Linux (CentOS, Ubuntu, RedHat, SUSE), FreeBSD. На практике этого набора достаточно. Однако, по мере необходимости, его можно пополнять и другими дистрибутивами. Современные дистрибутивы открытых операционных систем устанавливаются не сложнее ОС Windows, а иногда и превосходят их в простоте и скорости установки. Создать виртуальную машину на основе открытой ОС в виртуальной компьютерной лаборатории студенту не составляет никакого труда. Даже без специальной подготовки, при

**Сравнительные характеристики традиционного учебного процесса и учебного процесса, автоматизируемого с помощью виртуальной лаборатории**

Характеристики	Традиционная система образования	Виртуальная компьютерная лаборатория, интегрированная в учебный процесс
Возможность удаленной работы	Отсутствует возможность полноценной удаленной работы (для выполнения лабораторных и курсовых работ необходимо посещать университетские классы во внеучебное время)	Возможность выполнения внеаудиторных заданий и лабораторных практикумов в любое время и в любом месте (для работы необходим ПК с доступом в сеть Интернет)
Платформа для внеаудиторных работ	Для внеаудиторного изучения корпоративных информационных систем и развертывания клиент-серверных приложений учащемуся нужна мощная рабочая станция (ПК повышенной производительности)	Для внеаудиторного изучения корпоративных информационных систем и клиент-серверных приложений учащемуся нужна рабочая станция с выходом в сеть Интернет. Развертывание подобного ПО на клиентском ПК не требуется
Стоимость лицензий	Стоимость программной лицензии при самостоятельном изучении, которая необходима лишь на период обучения, оказывается слишком высокой для отдельно взятого пользователя	Лицензионное ПО уже предустановлено и доставляется на клиентские оконечные устройства (SaaS)
Надежность хранения данных	Надежность системы хранения данных и результатов работы на рабочей станции учащегося относительно невысока	Виртуальная компьютерная лаборатория основана на высоконадежном ЦОД с системой мониторинга, резервированного питания, удаленного конфигурирования, мгновенной перенастройки и восстановления данных
Доступность учебных материалов	Поиск учащимся лекционных, семинарских материалов, посещение библиотеки	Все необходимые дидактические материалы: видеоуроки, лекции, примеры работ и пр., размещены на интерактивном информационном интернет-ориентированном портале
Ограничение выбора платформы	Привязка учебного процесса к существующим программным платформам и продуктам	Возможность легкого выбора программной платформы для самостоятельного изучения информационных систем и технологий
Стоимость реинжиниринга	Требуется периодическая модернизация компьютерного парка университета	Нужна модернизация подсистем ЦОД и замена blade-серверов на более мощные, что обходится значительно дешевле

помощи интерактивного конструктора и подсказок, такая машина будет создана за короткий промежуток времени, сравнимый с установкой ОС Windows на домашнюю или офисную рабочую станцию. Последующая установка и настройка прикладного программного обеспечения на такую виртуальную машину не отличается от аналогичных действий в реальной машине. Отличительной особенностью является то, что виртуальная компьютерная лаборатория позволяет осуществлять доступ к виртуальной машине удаленно, посредством интернет-браузера. Важно отметить, что любая современная операционная система интересна для работы в такой лаборатории.

Использование открытых операционных систем семейства Unix/Linux позволяет применять находящиеся в свободном доступе средства разработки программ, аналогичные таким же средствам разработки в ОС Windows. К их числу относятся: NetBeans IDE; Gambas; Lazarus; Geany; альтернативные Windows-приложения средства Microsoft Visual Studio Express Edition; Embarcadero Technologies C++ и Delphi [6].

Внедрение представленной в настоящей статье виртуальной лаборатории, изучение с ее помощью материалов спецкурсов и освоение программ позволяют учащимся накапливать необходимый практический опыт для отраслевых знаний, для работы с виртуальными ресурсами вплоть до создания частных "облачных" решений на предприятии. Перечисленные знания, опыт и практические навыки значительно повышают конкурентоспособность выпускников, которых

готовят выпускающие кафедры Института системного анализа и управления университета "Дубна".

В настоящее время работающий образец виртуальной компьютерной лаборатории в университете "Дубна" успешно эксплуатируется в рамках курсов "Информационные системы в административном управлении", "Технологии разработки интернет-порталов и систем электронной коммерции", "Бизнес-реинжиниринг", "Информационные системы в бизнесе", "Современные подходы системного анализа".

**Виртуальный компьютерный класс как компонент виртуальной компьютерной лаборатории и его роль в учебном процессе университета "Дубна"**

*Виртуальный компьютерный класс* — это комплекс программно-аппаратных средств, основанный на технологиях виртуализации. Он позволяет проводить дистанционное обучение и осуществлять комплексную доставку приложений учащимся. Основой виртуального компьютерного класса является программная система Citrix XenApp™. Она представляет средства доставки приложений Windows, в этой системе приложения управляются из Центра обработки данных (ЦОД) и доставляются по запросу пользователя на любое клиентское конечное устройство в виде отображения экрана. Необходимо отметить следующие особенности виртуального компьютерного класса, организованного на основе Citrix XenApp™.

• Встроенные средства сканирования конечных систем и средства управления политиками использования ресурсов позволяют определить уровень доступа к ним и наилучший метод доставки приложений на основании таких атрибутов, как роль пользователя, характеристики устройства и состояние сети. Учет этих факторов обеспечивает доступность и гибкость применения виртуального класса пользователями.

• Реализован подход "один пароль и одна точка входа" для доступа к приложениям и данным. Это позволяет предупредить преднамеренное повреждение данных и фальсификацию работ.

• По сети передаются только движения мыши, нажатия клавиш и обновления экрана, в то время как все вычисления проходят в ЦОД. Это позволяет студентам работать с ресурсоемкими приложениями в виртуальном компьютерном классе на маломощной рабочей станции.

• Наличие единого пользовательского интерфейса и единого входа в систему обеспечивают высокий уровень удобства пользователю.

Виртуальный компьютерный класс представлен одной из машин виртуальной компьютерной лаборатории, к которой предоставлен доступ из внешней сети. Благодаря Citrix XenApp™ можно использовать единый эталонный образ каждого приложения и предоставлять его для использования в режиме online или offline. Удобство работы пользователей с доставленными таким образом приложениями выше, чем у приложений, развернутых на устройстве пользователя. Возникновение конфликтов с другим ПО (например, мультимедийным) в таком случае сведено к минимуму [7].

Благодаря успешному внедрению виртуального компьютерного класса в учебный процесс в качестве одного из ключевых компонентов виртуальной компьютерной лаборатории у университета появилась возможность доставлять как проприетарные программные продукты, так и свободно распространяемое ПО на компьютеры сотрудников университета и учащихся методом SaaS (*Software as a Service*). Основное преимущество SaaS состоит в отсутствии затрат, связанных с установкой, обновлением и поддержкой работоспособности оборудования и работающего на нем программного обеспечения. Такой подход (способ предоставления услуги) приспособливает приложения для удаленного использования, позволяя работать с ним нескольким клиентам. Подобно виртуальным машинам, развернутым в виртуальной компьютерной лаборатории, в Citrix XenApp™ возможно применять как проприетарное, так и свободно распространяемое ПО. Вместе с тем существует ограничение, связанное с обязательной совместимостью таких приложений с ОС Windows. Однако доставка этих приложений может проводиться и на другие программные платформы.

Опыт показывает, что изучение основ офисных информационных технологий, составление расчетов и графиков в табличных редакторах, создание презентаций, блок-схем, оформление отчетов, курсовых и дипломных работ на настоящее время нереализуемы на практике без комплекса офисных программ. Электронные таблицы, входящие в состав многих офисных пакетов, можно использовать не только для оформления документов, но и для проведения различных вы-

числений и исследований. В виртуальном компьютерном классе установлен знакомый всем проприетарный программный пакет Microsoft Office, а также кроссплатформенный свободно распространяемый офисный пакет программ IBM Lotus Symphony.

В качестве графического векторного редактора, предоставляемого студентам и преподавателям, работающим в виртуальном компьютерном классе, установлен проприетарный CorelDraw. Вместе с тем нет никаких ограничений при необходимости перейти на полнофункциональные кроссплатформенные свободно распространяемые GNU Image Manipulation Program (или GIMP), Inkscape и OpenOffice.org Draw [7].

Отдельно следует остановиться и на прикладном программном обеспечении, которое применяется в учебном процессе Института системного анализа и управления университета "Дубна", а именно — пакетах Maple, Mathcad, Mathematica, MATLAB. Студенты начинают знакомство с ними на базовых и специальных курсах, применяют эти пакеты в самостоятельных и лабораторных работах, курсовых и дипломных проектах. Такими же программами сотрудники университета пользуются в своей научно-исследовательской деятельности и практической работе. Существует достаточно большое число полнофункциональных и свободно распространяемых прикладных пакетов (Axiom, Maxima, Sage), отвечающих требованиям учебного процесса и тематике исследований, проводимых в университете. Любой из перечисленных выше программных пакетов может быть установлен для дальнейшей трансляции и доставки через виртуальный компьютерный класс на конечное пользовательское устройство.

Виртуальный компьютерный класс предоставляет более упрощенную схему работы для пользователей. При этом не возникает необходимости создавать виртуальные машины, как это делается в виртуальной компьютерной лаборатории. Не нужно устанавливать операционные системы и прикладное программное обеспечение на них, сталкиваясь с возможными ошибками и вопросами совместимости при установке таких приложений на рабочие персональные компьютеры.

Рассматриваемый виртуальный компьютерный класс позволяет предоставить учащимся удаленный доступ к необходимым программным и вычислительным ресурсам для выполнения заданий, связанных с освоением различных информационных систем и обеспечением доступа к базовым приложениям, находящимся в университете, независимо от местоположения учащегося. Кроме этого, ресурсы виртуального класса позволяют студентам выполнять научно-исследовательские и лабораторные работы, вести курсовые и дипломные проекты. По сути, такой класс представляет собой виртуальный аналог привычного нам компьютерного класса. Однако он обладает неоспоримыми достоинствами, предоставляющими университету возможность организовать учебный процесс для студентов любой формы обучения, основываясь на одной, а не на десятках приобретенных лицензий для программного продукта, не требуя обновления компьютерного парка в университете и наличия мощных персональных рабочих станций у сотрудников университета и студентов.

## Дальнейшие перспективы IT-образования

Основные принципы IT-образования в университете "Дубна" включают расширение направлений IT-подготовки и совершенствование программ учебных курсов и методов для интеграции современных информационных и образовательных технологий для повышения качества и доступности образования.

Тесное сотрудничество университета с Академическим Центром Компетенции IBM и рядом других IT-компаний, заинтересованных в высококвалифицированных кадрах, обеспечивают качественно иной уровень подготовки специалистов, повышение их конкурентоспособности и востребованности на рынке труда за счет создания оптимальных в образовательном плане условий и программ, нацеленных на обучение студентов и целевых групп.

Опыт эксплуатации виртуальной компьютерной лаборатории на основе технологии облачных вычислений и виртуального компьютерного класса в учебном процессе в течение двух лет, а также анализ эффективности, проведенный, в том числе, и на основании опроса выпускников Института системного анализа и управления 2009—2010 гг., показали, что знания, умения и навыки, полученные в рамках инновационного учебного процесса, удовлетворяют всем требованиям ведущих компаний отрасли. Учащиеся смогли более эффективно не только проходить устные собеседования, но и демонстрировать свою высокую квалификацию на практике, решая реальные задачи, принимая участие в реальных проектах. За счет увеличения потенциала выпускников повысилась их востребованность на рынке труда, что позволило увеличить число трудоустроенных выпускников в ведущие мировые компании.

Подводя итоги, важно отметить, что основными концептуальными положениями развития IT-образования в университете "Дубна" будут:

- "образование через всю жизнь" — установка на умение постоянно учиться;
- "образование через пространство и время" — идея реализации удаленного, индивидуального обучения, доступного в любое время и в любой точке нашей планеты;
- "свобода действий и свобода выбора" — внедрение в учебный процесс различных свободно распространяемых и проприетарных программных продуктов и систем, интеграция новых платформ и инновационных технологий;
- построение распределенной асинхронной среды обучения на базе социальной сети и входящего в ее состав расширенного набора социальных инструментов (ленты активностей, блогов, форумов, электронных библиотек и т. д.);
- способность к поиску новых знаний для достижения поставленной цели;
- развитие навыков работы в командах специалистов различных областей знания;
- постоянное творческое самосовершенствование.

В целях реализации этих концептуальных положений учебный процесс в университете "Дубна" развивается в следующих направлениях:

- выполняются программы саморазвития и самосовершенствования университета за счет появления новых подразделений;

- развиваются проектно- и проблемно-ориентированные методы обучения, междисциплинарные подходы к обучению;

- часть образовательных программ выносятся за пределы обязательных курсов и передается студентам для самостоятельного освоения с помощью современных компьютерных средств обучения и программно-технологической платформы виртуальной компьютерной лаборатории;

- создаются новые методические пособия, интерактивные лекции, предложения поисковых работ, инструкции, размещенные в виртуальной компьютерной лаборатории и доступные удаленно;

- создаются системы обеспечения качества подготовки специалистов, системы тестирования, контроля, методы оценки уровня усвоения знаний;

- создаются новые мотивационные установки для студентов, преподавателей, научных организаций и бизнеса, способствующие успешной деятельности университета.

Все эти процессы происходят в условиях быстро развивающейся глобализации образования, развития систем дистанционного обучения, появления новых программно-технологических и аппаратных платформ, массового использования ресурсов сети Интернет в целях образования.

## Список литературы

1. Белов М. А., Антипов О. Е. Принципы проектирования виртуальной компьютерной лаборатории на основе технологии облачных вычислений // Сб. трудов Международной конференции "Современные проблемы и пути их решения в науке, транспорте, производстве и образовании" 2010". Одесса: УкрНИИМФ, 2010.
2. Добрынин В. Н., Мазный Г. Л., Черемисина Е. Н. Концепция опережающего применения информационных систем в учебном процессе // Компьютерные технологии в образовании: научный семинар на IV съезде Российского союза ректоров высших учебных заведений. М.: Изд-во МГУ. 1996.
3. Черемисина Е. Н., Крейдер О. А. Инновационная практика подготовки IT-специалистов в университете "Дубна" // Сб. научных трудов каф. системного анализа и управления / под ред. Е. Н. Черемисиной. Вып. 2. М.: ООО "Центр информационных технологий в природопользовании", 2008.
4. Белов М. А., Антипов О. Е. Разработка и внедрение программно-аппаратной платформы виртуальной компьютерной лаборатории в образовательный процесс высшей школы. // Сб. трудов VII Международной научно-практической конференции "Наука и современность — 2010". Новосибирск: ЦРНС, 2010.
5. Menken I. Virtualization Architecture, Adoption and Monetization of Virtualization Projects using Best Practice Service Strategy, Service Design, Service Transition, and Continual Service Improvement Processes. London: Emereo Pty Ltd., 2008. 164 p.
6. Белов М. А., Антипов О. Е. Опыт использования открытого программного обеспечения в виртуальной компьютерной лаборатории на основе технологии облачных вычислений // Сб. трудов VI Международной научно-практической конференции "Проблемы и перспективы развития образования в России". Новосибирск: ЦРНС, 2010.
7. Антипов О. Е., Белов М. А., Бугров А. Н. Создание и опыт эксплуатации виртуального компьютерного класса как компонента виртуальной компьютерной лаборатории, основанной на ЦОД с применением серверов лезвийной архитектуры // Сб. трудов IV Международной научно-практической конференции "Наука в современном мире". М.: Спутник+, 2010.
8. Сорокин А. В. Использование облачных технологий в образовании // Труды II Международной конференции "Информационно-инновационные технологии: Интеграция науки, образования и бизнеса", Казахский национальный технический университет имени К. И. Сатпаева. Алматы, 2011.

## Посткраудсорсинг как архитектура экспертных сетей

*Анализируются посткраудсорсинговые технологии построения экспертных сетей и их место в общем ряду информационных технологий. Показано отличие архитектуры экспертных сетей от краудсорсинговых проектов. Экспертные сети представлены как эффективный инструмент коллективной работы в условиях информационного "взрыва", вызванного вхождением общества в информационную эпоху. Рассмотрены различные пути развития посткраудсорсинговых технологий.*

**Ключевые слова:** краудсорсинг, социальные сети, экспертные сети, автоматизация, информационный "взрыв", информационное общество

### Введение

Программные продукты (ПП) эволюционируют согласно спросу на информационные технологии (ИТ). В начале своего развития такой спрос в основном создавали наука и оборонная отрасль, используя ПП для численного моделирования. Позже ПП взяли на вооружение крупные корпорации для планирования и учета, а потом уже и весь бизнес. С конца прошлого века ПП стали массово использоваться и для частной жизни, приведя к созданию целой отрасли онлайн-сервисов: поисковых систем, интернет-коммуникаций, социальных сетей и др. Вообще говоря, ИТ имеют как техническую, так и социальную сторону, поскольку любые технологии являются инструментом деятельности человека, определяют возможности управления окружающей его средой. И именно социальная сущность технологий позволяет их классифицировать и определять направление их развития. Основой одной из таких социальных классификаций может служить отношение к использованию человеческих ресурсов (*sourcing*). В частности, можно выделить следующие формы систем управления: *insourcing*, *outsourcing* и *crowdsourcing*, которые появляются в разные эпохи и формируют структуру экономики, словно годовые кольца дерева [1].

*Инсорсинг* стал символом индустриализации мировой экономики, поскольку предполагает глубокую специализацию внутри предприятия. Если ранее профессия человека соответствовала, вплоть до совпадения в названии, продукту или услуге, которую он производил (сапожник, ювелир, музыкант, цирюльник и т. п.), то в индустриальную эпоху профессией стали

отдельные производственные функции (слесарь, токарь, бухгалтер, юрист и т. д.). Инсорсинг как инструмент управления лежит в основе всех современных архитектур корпоративных информационных систем (MRP, ERP, CRM и т. п.).

Следующая по времени эра — постиндустриальная, она уже связана с технологией *аутсорсинга*. На основе новых профессий, возникших в предыдущую эпоху, начали возникать многочисленные и высокотехнологичные предприятия, оказывающие монофункциональные услуги компаниям — производителям товаров. Производство товаров рассыпалось на множество промежуточных сервисов, образовав новую глобальную экономику услуг и систему взаимосвязанных предприятий.

Наступление новой информационной эпохи породило и новую технологию (*краудсорсинг*), позволяющую привлекать в качестве ресурсов глобальные сообщества людей (*crowd* — толпа). Краудсорсинг получил распространение благодаря глобальным сетевым технологиям, предоставившим сразу многим людям доступ к общим информационным ресурсам и возможность участвовать в массовой коллективной деятельности, не ограничиваясь одной территорией. Краудсорсинг пользуется все большим и большим вниманием как со стороны бизнеса, так и со стороны государства.

Однако и у этой, самой современной в настоящее время, технологии уже появляются "наследники", которые будут востребованы в следующей за информационным обществом эпохе — в обществе знаний. Новый посткраудсорсинговый этап развития техно-

логий, описанный в настоящей статье, предполагает использование ресурсов самоорганизующихся профессиональных сетевых сообществ. Именно посткраудсорсинговые технологии призваны обуздать информационный "взрыв", вызванный массовым доступом населения к сетевым системам публикации.

## Краудсорсинг и его ограничения

Однозначных и общепризнанных определений термина краудсорсинга нет. В подробном диссертационном исследовании, посвященном краудсорсингу, С. Гирц определяет новую технологию как "онлайн-аутсорсинг задач, выполняемый частными лицами (или группой лиц) при условии свободного их доступа к участию" ([2], стр. 4)\*. Это противоречит мнению автора термина Д. Хауи, который в своей знаменитой статье [3] приводит слова вице-президента по инновациям компании Procter & Gamble (P & G) Л. Хьюстона: "Люди ошибаются, считая эту технологию аутсорсингом, она ею не является. Аутсорсинг — это, когда я нанимаю кого-то для оказания услуги, они выполняют работу, и отношения на этом заканчиваются. Это не сильно отличается от обычного найма, известного на протяжении веков. Мы же говорим о привлечении людей извне и вовлечении их в широкий творческий и совместный процесс. Это совершенно новая парадигма".

Стоит согласиться с топ-менеджером компании P & G, что одним из кардинальных отличий краудсорсинга от аутсорсинга является отсутствие каких-либо юридически значимых соглашений между инициатором задачи и участниками ее решения: все строится на энтузиазме и естественном желании человека участвовать в творческом процессе. Впрочем, и Гирц соглашается, что именно неопределенность конкретных исполнителей является одной из трех дефиниций краудсорсинга (наряду с наличием заказчика и свободным доступом к участию). Учитывая, что краудсорсинг поставлен в один ряд с такими понятиями, как инсорсинг и аутсорсинг, являющимися своего рода инфраструктурными политиками своих эпох, целесообразно дать следующее определение краудсорсингу. *Краудсорсинг — это технология привлечения всего населения к решению общезначимых и творческих задач, включая контроль над деятельностью бизнеса и органов власти, на основе энтузиазма, добровольности и равноправного доступа к информации и участию.* Именно такое определение делает краудсорсинг не только технологией решения задач, но и идеологией информационного общества, в котором информационная прозрачность всей деятельности предприятий, организаций и органов власти становится обязательной.

Заметим, что приведенному выше определению удовлетворяет, например, задача выборов в органы государственного управления в демократических

\* "...online outsourcing of a task to (a group of) private individuals in the form of an open call".

странах, и в этом смысле технология краудсорсинга отнюдь не молода. Но только в последние годы, когда средства коммуникаций и компьютеры оказались доступны всему населению, эта технология стала играть существенную роль в жизни общества. Появились и многочисленные варианты ее использования. В зависимости от типа задач и участников их решения, автор термина "краудсорсинг" Д. Хауи выделил четыре таких варианта: *crowd wisdom* (коллективное мышление), *crowd creation* (коллективное созидание), *crowd voting* (опросы и голосования) и *crowdfunding* (коллективное финансирование), первый из которых Гирц, в свою очередь, разделил еще на три: *prediction markets* (коллективное предсказание), *crowdcasting* (коллективное вещание) и *brainstorming* (мозговой штурм). Выделение технологии *brainstorming* в отдельный тип связано с тем, что именно задачи в области коллективного поиска новых идей (*idea management*) стали одним из новых направлений в области корпоративной автоматизации.

В табл. 1 представлены отличия разных моделей сорсинга. Указание инновационности как основной задачи краудсорсинга может вызвать протест, поскольку инновации с давних времен являются прерогативой науки. Ниже в статье будет показано, что развитие научного сообщества и краудсорсинговых технологий в информационном обществе движутся в одном направлении. Кроме того, в условиях расширения численности научного и инновационного сообществ возникает эффект информационного "взрыва", требующий новых технологий работы с информацией.

Несмотря на популярность краудсорсинга, имеются и недостатки этой технологии, которые существенно ограничивают область ее применения. Во-первых, открытость доступа людей к решению задач (*open call*) порождает массу непрофессиональной активности, отпугивающую настоящих специалистов. Во-вторых, в силу добровольного (необязательного) характера участия краудсорсинг подходит только для решения тех задач, которые просты и вызывают интерес. Необходимые же, но незаметные или сложные задачи оказываются за бортом технологии. Краудсорсинг эффективен, например, в качестве инструмента контроля над примитивными государственными закупками, но неприменим в роли общественной экспертизы сложных и наукоемких проектов. Именно посткраудсорсинговые технологии призваны преодолеть указанные ограничения использования ИТ в деятельности человека.

## Три направления автоматизации жизнедеятельности человека

Во избежание фетиша самих технологий (которым порой присваиваются демонические свойства самих социальных отношений), целесообразно любые инновации в информационно-коммуникационной области рассматривать как процесс автоматизации деятельности человека. Это очевидно в случае

Сравнение различных моделей сорсинга

Характеристики сорсинга	Инсорсинг	Аутсорсинг	Краудсорсинг
Используемые ресурсы	Сотрудники компании	Партнеры компании	Окружение и сотрудники компании
Выполняемые задачи	Автоматизация функций	Совершенствование функций	Инновационное развитие
Информационные системы	ERP и цеховые системы	B2B-системы	Idea management
Инициаторы решения задач	Компания	Компания или партнеры	Компании, население
Общее значение для экономики	Разделение труда по функциям	Отраслевое развитие	Вовлечение населения

Сравнительные характеристики направлений автоматизации

Особенности автоматизации	Корпоративная автоматизация	Автоматизация личной жизни	Экспертные сообщества
Роль человека в решении задачи	Полностью определена информационной системой	Полная свобода действий	Определена опытом работы
Решаемая задача	Повышение эффективности бизнеса	Повышение качества жизни	Профессионализм
Объединение людей	В рамках производства услуги	В рамках общих интересов	В рамках одной профессии
Информационные системы	ERP, B2B, MES, CRM-системы	Социальные сети	Экспертные сети
Общественное назначение	Прозрачный бизнес	Свободный доступ к информации	Коллективное знание

информатизации работы предприятий, где автоматизированные системы направлены на обеспечение конкретной функциональной деятельности подразделений: финансовый и кадровый учет, поддержка продаж, цеховая автоматизация, формирование планов и бюджетов и т. д. Но такой "внедренческий" подход совсем не очевиден в бытовой практике, поскольку освоение новых возможностей человеком в личной жизни проходит незаметно и без видимого принуждения. Пожилые люди изучают Skype, чтобы общаться со своими внуками, которые уехали за границу. Молодежь делится опытом друг с другом об удобствах обмена файлами через социальные сети, люди среднего возраста реализуют свои потребности быть в центре внимания через блогосферу и Twitter. Но все эти непохожие друг на друга инструменты представляют собой не что иное, как автоматизацию деятельности человека.

Процесс глобальной автоматизации жизнедеятельности человечества идет по трем направлениям. Первое — это информатизация бизнеса и государственной деятельности. Основой этого направления является индустрия корпоративной автоматизации, бурно развивающаяся в последние два десятилетия. Второе направление — это создание новой коммуникационной среды досуга человека, которая расширяет возможности общения с друзьями, бывшими коллегами и соучениками, более полно удовлетворяет потребности в информации и в самовыражении, позволяет существенно увеличить возможности самообслуживания. Это направление получило широкое развитие

в последние годы в рамках распространения социальных сетей и облачных технологий массового потребления.

И наконец, третье направление — это экспертные сети, которые объединяют людей по профессиональному признаку и позволяют формировать и поддерживать сетевые сообщества, необходимые как для реализации глобальных и инновационных проектов, так и для роста квалификации участников таких сетей. Это направление автоматизации строится на технологии посткраудсорсинга и пока еще ждет своего пика развития.

В табл. 2 приведены сравнительные характеристики трех направлений автоматизации. Сетевые экспертные сообщества станут востребованными по мере автоматизации и интеллектуализации основных производственных функций на предприятиях. Профессиональная деятельность сотрудников компаний, значительно продвинувшихся в области автоматизации, все больше будет инновационной и экспертной. Авторы Викиномики Д. Тапскотт и Э. Д. Уильямс так описали эту ситуацию [4]: "Создание гибких самоорганизующихся команд, предназначенных для решения одной или нескольких конкретных задач, скоро превратится в норму. Консультирование станет основной моделью работы, и можно ожидать, что все больше сотрудников начнут требовать за свой интеллектуальный вклад долю в прибыли компании". Архитектура экспертных сетевых сообществ имеет общие черты как социальных сетей, так и корпоративных информационных систем.

## Архитектура экспертных сетей

Экспертная сеть — это сообщество профессионалов в определенной области, объединенных одной онлайн-платформой, в рамках которой они обмениваются информацией, оценивают работы друг друга и иные работы по тематике своей профессии. Сетевая платформа позволяет специалистам коммуницировать (как в социальной сети), а также четко регламентировать процессы своего взаимодействия подобно корпоративным системам. Экспертная сеть является инструментом, организующим работу множества отдельных экспертов по принципу коллективного разума, и призвана выполнять задачи, решение которых традиционным способом (при помощи формальных проектных групп в компаниях или в рамках краудсорсинга) невозможно или неэффективно. Примерами таких задач могут быть проведение общественной экспертизы сложных проектов, решение инновационных и изобретательских задач, выработка стандартов и методик и т. п.

Экспертная сеть — логическое развитие концепции краудсорсинга, но с более высокими требованиями к уровню профессионализма участников и с элементами обязательности по участию. Пользователи социальных сетей не берут на себя никаких обязательств. Например, люди могут иметь профили в LinkedIn и даже могут поддерживать его в актуальном состоянии (при смене работы), но совсем не обязаны участвовать в деятельности какого-либо сообщества в сети. Напротив, в экспертной сети каждый участник принимает на себя обязательства по участию в работе сети и выполняет их. Разумеется, обязательства не должны быть обременительными, а выполнение их должно нести явную выгоду для конкретного эксперта.

Экспертные сети во многом напоминают сетевые сообщества, которые реализуют какой-либо проект (например, в области открытого программного обеспечения). В сетевых проектах также важен рейтинг и профессионализм участников, они совместно работают над общими документами или программными кодами. Однако такие сетевые сообщества монопроектны и не призваны явно развивать профессионализм своих участников, совместно перерабатывать информацию. Их деятельность, как правило, финансируется какими-либо компаниями или спонсорами, и работа менеджеров таких сетевых проектов соответствует скорее корпоративной деятельности, нежели краудсорсинговой технологии.

Архитектура экспертных сетей наиболее близка схеме организации науки. Научное сообщество осуществляет взаимную оценку работ через экспертизу публикаций, оно имеет свои рейтинги (научные степени, индекс цитируемости, академические звания и т. д.), основанные на оценках коллег, и инструменты для профессионального обмена информацией — научные журналы. В некотором смысле экспертные сети можно считать системой автоматизации для научной деятельности [5], которая, правда, пока еще слабо внедряется в жизнь научного сообщества, несмотря

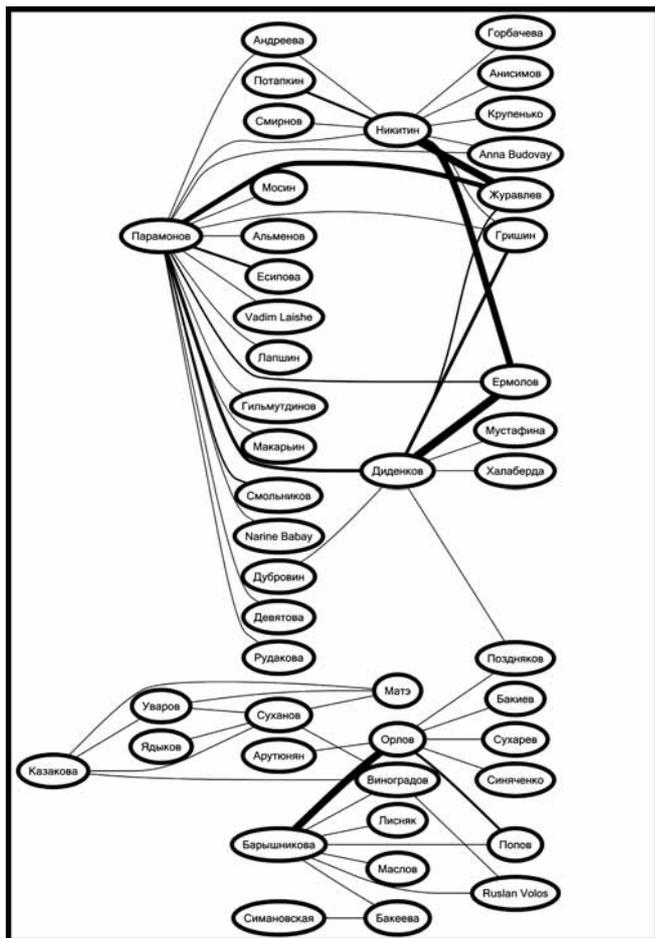
на провозглашенное Б. Шнейдерманом пришествие технологий web 2.0 в науку [6]. Однако уже в недалеком будущем автоматизация деятельности научного сообщества на основе посткраудсорсинговых технологий станет неизбежной. Это связано с информационным "взрывом" в обществе, начало которого положено ростом числа людей, занятых в инновационной и творческой отраслях, и существенным упрощением электронных публикаций.

## Посткраудсорсинг и информационный "взрыв"

Информационный "взрыв" легко проиллюстрировать на простой модели. Предположим, что средний ученый (инноватор, методист и т. п.) регулярно публикует шесть статей в год. Пусть при этом он ежедневно читает десять статей своих коллег (около 250 в месяц). Если все активны в той же мере (т. е. публикуют одну статью каждые два месяца), легко получить, что число коллег, все статьи которых читает наш ученый или инноватор, будет не более 500. Если число специалистов в данной отрасли превышает эту цифру, часть информации наш ученый физически прочитать уже не успеет. В доинформационные времена число ученых или специалистов, работающих в инновационных сферах, было невелико, и каждый имел информацию о деятельности всех своих коллег. Информационная эпоха нарушила это равновесие.

Противостоять информационному "взрыву" возможно только коллективно. На самом деле так и происходит сегодня в научном сообществе. Ученые ориентируются на рекомендации своих коллег, сужая таким образом количество необходимой для анализа информации. В общей массе все статьи читаются, но не все — каждым. Именно коллективное потребление информации наряду с производством новых знаний и является одной из основных задач посткраудсорсинговых технологий. В экспертной сети анализируется каждая новая информация по теме, но каждый раз небольшим числом выбранных по алгоритму экспертов, которые в отличие от коллег по данной публикации обязаны представить рецензии и рекомендации. Экспертиза ограничивает "обязательное" для всех чтение лишь выдающимися статьями и проектами, заменяя остальные публикации рецензиями или дайджестами. Экспертные сети становятся реальными коллективными "мозговыми" центрами, позволяя обрабатывать информацию, превышающую возможности одного человека.

Интеллектуальные алгоритмы экспертных сетевых сообществ (в отличие от социальных сетей) являются основой их работы. Среди реализаций таких алгоритмов стоит отметить системы поиска экспертов, позволяющие найти специалиста посредством анализа близости информационного контента его деятельности к искомой компетенции. К таким же инструментам следует отнести и системы поиска информационных разрывов, т. е. поиска отсутствия коммуникаций меж-



Иерархия и коммуникации экспертов в области одной компетенции

ду группами экспертов, которые занимаются общими вопросами (имеют одинаковые компетенции [7]. На рисунке изображен один из результатов поиска таких несвязанных групп экспертов внутри одного предприятия (поиск осуществлялся для одной компетенции), толщина линии показывает объем коммуникаций. Нетрудно видеть, что на анализируемом предприятии есть две группы экспертов из одной области компетенций, имеющие слабые коммуникации между собой.

Несмотря на отсутствие организационной иерархии в экспертных сетях, в них все время проходит оценка участников, позволяющая определить их профессионализм в определенных компетенциях. Данные рейтинги позволяют алгоритмически распределять активности членов, назначая последним определенную роль в экспертизах или проектах. В отличие от типичных рабочих групп, сеть экспертов масштабируема и может состоять из тысяч и десятков тысяч участников, что позволяет оперативно привлекать для решения задачи нужных специалистов.

Эксперты принимают добровольное участие в работе сети, но они имеют и обязательства по проведе-

нию экспертиз. Выгода участников сети состоит как в материальном вознаграждении, так и в росте личного профессионализма (за счет коллективного анализа информации) и в повышении своего авторитета, что ведет, в конечном счете, к увеличению стоимости на рынке труда. Оплата активности (прямая или опосредованная) наряду с обязательностью участия отличает экспертные сети от краудсорсинговых проектов. Экспертные сети пока еще не заняли своего места в социальной организации людей, занимающихся инновациями и наукой, но за ними будущее. Недаром авторы "Викиномики" (см. [4]) писали: "Одно из крупнейших изменений в следующем десятилетии будет связано с переходом от добровольного участия в пиринговых сообществах, не связанного с финансовым вознаграждением, к модели, участники которой способны автоматически рассчитать финансовый эффект своего участия". Впрочем, в экспертной сети всегда будут экспертизы, стоимость которых невысока (или даже равна нулю), но участие в которых престижно или интересно — как в краудсорсинговых проектах.

### Перспективы развития посткраудсорсинговых технологий

Краудсорсинговые технологии привлекают бизнес не столько своей инновационностью, сколько масштабом привлечения населения и возможностью публикации рекламы. Если в качестве примера рассмотреть инициативу Сбербанка России по коллективной выработке долгосрочной стратегии своего развития с привлечением краудсорсинга (осенью 2011 г.), трудно определить, сколько в такой акции от реальной задачи по созданию стратегического документа, а сколько от рекламы нового технологичного имиджа финансового учреждения. Краудсорсинг существенно расширил рекламные площадки в интернете, которые до этого успешно были освоены поисковыми и коммуникационными сервисами.

Технология социальных сетей — одна из основных отправных точек развития посткраудсорсинга. По мере того как универсальный функционал, необходимый для удовлетворения потребности в социализации всего населения, полностью сформируется, социальные сети будут предлагать ориентированные на более узкие группы инструменты. Уже сейчас некоторые из социальных сетей предоставляют услуги профессиональным сообществам, например [ted.com](http://ted.com), [linkedin.com](http://linkedin.com), [biznik.com](http://biznik.com), [ecademy.com](http://ecademy.com) на Западе, [e-xecutive.ru](http://e-xecutive.ru), [gosbook.ru](http://gosbook.ru), [4cs.ru](http://4cs.ru) в России. Однако сближению социальных сетей с экспертными мешает существенное различие в схемах финансирования. Если монетизация социальных сетей строится на рекламе и оказании платных услуг самим пользователям сети, то экспертные сети существуют за счет проведения экспертиз для бизнеса и государства, объем которых пока еще существенно меньше рекламного рынка. Именно в силу особенностей монетизации социальные сети

заинтересованы в увеличении числа своих пользователей, а экспертные сети — в росте профессионализма участников.

Еще одним путем развития посткраудсорсинговых технологий могут стать порталы профессиональных сообществ, которые от информационной модели постепенно переходят к сервисной. В отличие от социальных сетей, ресурсы профессиональных сообществ вызывают больше доверия у специалистов и при достаточном развитии функционала вполне могут стать удобным инструментом деятельности экспертной сети. Этот путь имеет хорошие перспективы еще и потому, что профессиональные сообщества уже часто оказывают экспертные услуги бизнесу и государству, подчас имеют собственные рейтинговые центры и информационные издания. Немалую роль существующие экспертные сообщества оказывают и на политику государств [8].

Третий путь развития краудсорсинга будет непосредственно связан с автоматизацией научной деятельности (программа Science 2.0), а также с обеспечением инструментарием экспертных групп, деятельность которых в инновационной области приобретает массовый характер во всем мире. В настоящее время за счет смещения требований рынка от расширения функциональности товаров к увеличению их интеллектуальности возросли возможности венчурного капитала, нацеленного на инновации. Одновременно, благодаря массовому расширению коммуникационных площадок, вырос и рынок предложений идей. От этих процессов явно отстают возможности существующих экспертных групп, призванных стать необходимым связующим звеном между новыми идеями и инвестициями. Экспертные сети могут стать крайне востребованным инструментом роста инноваций в силу своей масштабируемости и высокого профессионализма.

## Заключение

Краудсорсинговые технологии предоставляют населению удобные площадки для проявления собственной инициативы и помогают эти инициативы использовать для досуговых, общественно значимых или коммерческих целей. Посткраудсорсинговые технологии, напротив, направлены прежде всего на организацию самих сообществ и повышение квалификации их пользователей. Эти технологии необходимы для обеспечения коллективной деятельности специалистов, занимающихся профессиональной и инновационной деятельностью. В условиях информационного "взрыва" посткраудсорсинг становится

единственной технологией, позволяющей преодолеть информационную ограниченность индивида, и ведет к созданию "Коллективного Разума", о котором писал один из идеологов синергетики Н. Моисеев [9].

В этой связи представляется целесообразным введение отдельного термина для обозначения посткраудсорсинговых технологий — ноосорсинг (где "ноо" обозначает коллективный разум). Данный термин хорошо вписывается и в классификацию типов сорсинга (см. введение к настоящей статье): если краудсорсинг соответствует информационной эре, то *ноосорсинг* относится к следующей за ней эпохе знаний. Новые технологии, которые опережают свое время и не согласуются с социальными изменениями, не развиваются. И наоборот — общественные метаморфозы, даже опережая технологические возможности, способны превратить самые обычные коммуникационные инструменты в уникальные средства преобразования социальной среды.

*Данное исследование проводилось при финансовой поддержке Правительства Российской Федерации (Минобрнауки России) в рамках договора № 13.G25.31.0096 о "Создании высокотехнологичного производства кросс-платформенных систем обработки неструктурированной информации на основе свободного программного обеспечения для повышения эффективности управления инновационной деятельностью предприятия в современной России".*

## Список литературы

1. **Славин Б.** От краудсорсинга к ноосорсингу // Открытые системы. 2012. № 01. С. 60–62.
2. **Geerts S. A. M.** Discovering Crowdsourcing Theory: Classification and Directions for use. Eindhoven, 2009. 117 p.
3. **Howe J.** The Rise of Crowdsourcing. // Wired. June 2006. URL: <http://www.wired.com/wired/archive/14.06/crowds.html>
4. **Тапскотт Д., Уильямс Э. Д.** Викиномика. Как массовое сотрудничество изменяет все. СПб.: BestBusinessBooks, 2009. 392 с.
5. **Славин Б.** Ноосорсинг как технология формирования "Науки 2.0" / Современные информационные технологии и ИТ-образование / Сб. избранных докладов научно-практической конференции: учебно-методическое пособие. Под ред. В. А. Сухомлина. М.: ИНТУИТ.РУ, 2011. 1052 с.
6. **Shneiderman B.** Web Science: A Provocative Invitation to Computer Science // COMMUNICATIONS OF THE ACM. 2007. June. Vol. 50. N 6. P. 25–27.
7. **Romanov D., Sidorov P.** Project or Process? How to Measure the Real Type of Employees' Activity // International Symposium on Advances in Social Systems (ASS 2011). 12–14 December 2011, Sydney, Australia. IEEE CS Press, 2011. P. 1084–1088.
8. **Казакова Е.** Роль экспертных сообществ России в политической модернизации // Власть. 2011. № 03. С. 11–14.
9. **Моисеев Н.** Универсум. Информация. Общество. М.: Устойчивый мир, 2001. 200 с.

## CONTENTS

**Galatenko V. A., Kostukhin K. A., Shmyrev N. V., Aristov M. S.** Using Open Source Tools for Programs Source Code Static Analysis in the Process of Developing Applications for Real-Time Operating Systems . . . . . 2

Modern real-time operating systems (RTOS) have some characteristics related to the particular of inferior quality. In this paper typical RTOS developers errors discussed, as well as methods of diagnosis and detection of these errors based on the use of static analysis of source code.

**Keywords:** static analysis, debugging, RTOS

**Korzun D. G., Lomov A. A., Vanag P. I.** Automated Model-Driven Agent Development for Space-Based Applications on the Smart-M3 Platform. . . . . 6

The paper considers the problem of automated model-driven application development for Smart-M3 when structure of problem domain is specified as OWL-ontology. Each software agent uses its own OWL-ontology to access the smart spaces in accordance with its role and interests. Given the ontology, the code of ontological library is generated in a target programming language. Transformation of the high-level OWL-model of problem domain into the low-level RDF-model of Smart-M3 is hidden from the developer. Our solution is realized in SmartSlog, a high-level software development kit for Smart-M3.

**Keywords:** smart spaces, Smart-M3, OWL, RDF, ontology library, code generation

**Kostyuk V. V., Bovkunovich M. A.** Elements of Service-Oriented Development of Architecture for Software on Base of Business Models . . . . . 15

In this article are regarded questions of receiving of WSDL interface on base of principle "Top-Down", etc. from models to code by transformation of models of business level to code level. General target of transformation — to resolve of problem of decomposition of information system corresponding to business processes. This target is reaching. This corresponds to main conception SOA — confluence of business logic with logic of information technologies realizing this business logic.

**Keywords:** business modeling, service-oriented architecture, SOA, transformations, generation of code, WSDL

**Soloviev S. V., Zatoka I. V., Lavlinskaya O. P.** Particularity of Information Systems Projecting Used for Supporting Technical Information Protection Activity. . . 21

It is examined the possibility of expert methods usage by information systems projecting used for supporting technical information protection activity on the example of information systems in the sphere of personal data protection.

**Keywords:** personal data, personal data information system, expert system

**Zhukov I. U., Mikhaylov D. M., Kukushkin A. V., Stari-kovski A. V., Ivanova E. V., Fedorov E. D., Tolstaya A. M.** Program Methods of Recognition of the Gastrointestinal Tract Sections, Polyps and Bleedings in the Wireless Capsule Endoscopy. . . . . 28

This article is devoted to module description that increases the efficiency of pathologies and morphological changes detection in the gastrointestinal tract and its sections on the images from the wireless endoscopic capsule. The method of bleeding and sections detection of the gastrointestinal tract is based on pixel analysis. Detection of polyps is based on analysis of convexity of the surface. 3D-models are constructed and measured. The algorithm indicates that in the original image at the pixel corresponding to this element a polyp takes place.

**Keywords:** capsule endoscopy, bleeding, polyp, sections, brightness, saturation, pixel analysis of the curvature.

**Cheremisina E. N., Belov M. A., Antipov O. E., Sorokin A. V.** Innovative Approach to E-learning in Dubna University based on Virtual Computerized Laboratory and Cloud Computing . . . . . 34

The article describes best practices for training highly qualified specialists with use of virtual computer laboratory, based on service-oriented architecture and cloud computing technology. The authors focuses on key components of the virtual computer lab, its integration, and explain step by step information about hardware and software used for development of system kernel, as well as educational and methodological aspects of successful use in the learning environment.

**Keywords:** virtual computer lab, cloud computing, innovative technique of corporate software studying, training of highly qualified IT professionals

**Slavin B. B.** Postcrowdsourcing as a Architecture of Expert Networks. . . . . 42

The paper analyzes the architecture of the expert networks and their place in the overall spectrum of information technology. The author displays the functionality difference between network of experts and the social network. The paper points to correlation the appearance of post-crowdsourcing technology and the "information explosion". The different paths of development of postcrowdsourcing technology are described.

**Keywords:** crowdsourcing, social networks, expert networks, automation, "information explosion", information society

ООО "Издательство "Новые технологии". 107076, Москва, Стромьинский пер., 4

Дизайнер *Т.Н. Погорелова*. Технический редактор *Е.М. Патрушева*. Корректор

Сдано в набор 14.06.2012 г. Подписано в печать 26.06.2012 г. Формат 60×88 1/8.  
Цена свободная.

Оригинал-макет ООО "Авансд солюшнз". Отпечатано в ООО "Авансд солюшнз".  
105120, г. Москва, ул. Нижняя Сыромятническая, д. 5/7, стр. 2, офис 2.