

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ  
УНИВЕРСИТЕТ РАДИОЭЛЕКТРОНИКИ

ISSN 0135-1710

**АВТОМАТИЗИРОВАННЫЕ  
СИСТЕМЫ УПРАВЛЕНИЯ И  
ПРИБОРЫ АВТОМАТИКИ**

**Всеукраинский межведомственный  
научно-технический сборник**

**Основан в 1965 г.**

**Выпуск 173**

Харьков  
2015

## СОДЕРЖАНИЕ

<b>ЖМАЕВА Ю.В., УДОВЕНКО С.Г., ЧАЛАЯ Л.Э.</b> АДАПТИВНОЕ ПРОГНОЗИРОВАНИЕ ОПТИМАЛЬНОГО КОЛИЧЕСТВА РЕСУРСОВ IT ПРОЕКТА ПО МЕТОДОЛОГИИ AGILE.....	4
<b>ДУДИНОВА О.Б., УДОВЕНКО С.Г.</b> МЕТОД НЕЙРОСЕТЕВОЙ ОБРАБОТКИ ЗАШУМЛЕННЫХ ПРОСТРАНСТВЕННЫХ ДАННЫХ.....	14
<b>ЧАЙНИКОВ С.И., СОЛОДОВНИКОВ А.С.</b> ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ СИНТЕЗА СТРУКТУРЫ ПРОБЛЕМНО-ОРИЕНТИРОВАННЫХ ПРОГРАММНЫХ КОМПЛЕКСОВ.....	24
<b>ЧАЛАЯ Л.Э., ЧИЖЕВСКИЙ А.В.</b> МЕТОД АВТОМАТИЧЕСКОГО ПОСТРОЕНИЯ ОНТОЛОГИЧЕСКИХ МОДЕЛЕЙ С ДРЕВОВИДНОЙ СТРУКТУРОЙ КОНЦЕПТОВ.....	32
<b>КОЖУХІВСЬКИЙ А.Д., НАМОФІЛОВА О.О.</b> ЗАСТОСУВАННЯ ГЕНЕТИЧНИХ АЛГОРИТМІВ У ЗАДАЧІ ПРО УКЛАДАННЯ РАНЦЯ.....	43
<b>ТАМЕР ВАНІ АМЕР, ХАХАНОВ В.И., ЕМЕЛЬЯНОВ И.В., ЛЮБАРСКИЙ М.</b> КОМПЬЮТИНГОВЫЕ МОДЕЛИ ОБЛАЧНЫХ СЕРВИСОВ.....	48
<b>БАРАННИК В.В., ШУЛЬГИН С.С.</b> ЭФФЕКТИВНОЕ КОДИРОВАНИЕ СЛОТОВ Р-КАДРОВ В ПРОСТРАНСТВЕ СТРУКТУРНО-ЛОКАЛЬНЫХ ОГРАНИЧЕНИЙ ДЛЯ ПОВЫШЕНИЯ БЕЗОПАСНОСТИ ДИНАМИЧЕСКИХ ВИДЕОРЕСУРСОВ.....	58
<b>БАРАННИК Д.В., СТАСЕВ С.Ю., МЕДВЕДЕВ Д.О.</b> ОБОСНОВАНИЕ ПОДХОДА ДЛЯ ФОРМИРОВАНИЯ КВАНТОВАННОГО ОПИСАНИЯ ТРАНСФОРМАНТЫ СЕГМЕНТА АЭРОФОТОСНИМКА.....	63
<b>БАРАННИК В. В., ОКЛАДНОЙ Д.Е., СУПРУН О.В.</b> АНАЛИЗ ХАРАКТЕРИСТИК СИСТЕМ CDMA ОТНОСИТЕЛЬНО ИНТЕНСИВНОСТИ БИТОВОГО ПОТОКА.....	68
<b>РЕФЕРАТЫ.....</b>	75
<b>ПРАВИЛА ОФОРМЛЕНИЯ РУКОПИСЕЙ.....</b>	79

## АДАПТИВНОЕ ПРОГНОЗИРОВАНИЕ ОПТИМАЛЬНОГО КОЛИЧЕСТВА РЕСУРСОВ ИТ ПРОЕКТА ПО МЕТОДОЛОГИИ AGILE

---

Предлагается метод расчета и прогнозирования оптимального количества ресурсов ИТ проекта по методологии Agile. Данный метод разрабатывается для управления риск менеджментом при проектировании программного обеспечения. Осуществляется программная реализация метода на языке программирования Java с использованием Spring MVC Framework.

### 1. Введение и постановка задачи

Классические индикаторы эффективности разработки программного обеспечения связаны с утилизацией ресурсов, позволяющей сравнивать фактически выполненный и запланированный объем работ. Следует отметить, что такие индикаторы сложны в восприятии и применяются только в классических инструментах планирования (например, в методиках, основанных на использовании диаграмм Ганта).

В последнее время развиваются методы создания Agile проектов с применением более простых и наглядных индикаторов состояния процесса разработки (например, коэффициента возврата инвестиций, планируемых и затраченных ресурсов и т.п.). Однако локальное использование каждого из таких показателей не является достаточно информативным для динамического отображения степени эффективности проектирования. В то же время комплексное использование совокупности характеристик и метрик Agile позволяет прогнозировать скорость и качество выполнения проектных задач. Agile методологии, как правило, основаны на построении диаграммы выполнения задач, которая позволяет проанализировать существующие проблемы в момент разработки и устранить их для успешного завершения итерации. График прогресса разработки – широко используемый способ, с помощью которого команда отслеживает темп выполнения итерации. Концепция диаграммы выполнения задач состоит в том, чтобы в выбранной контрольной точке отобразить, как много задач осталось выполнить. Применение данного артефакта в сфере управления проектами дает возможность проанализировать узкие места на этапах планирования, разработки и поддержки приложения.

Существующий рынок информационных технологий предлагает различные системы отслеживания ошибок, оперирующие определенным набором характеристик для построения диаграмм производительности, совокупного потока, графиков прогресса разработки и др. [1, 2]. Однако даже самые современные системы предоставляют лишь часть необходимых инструментов для построения диаграммы выполнения задач (burn-down chart). Используемые здесь показатели (скорость команды, длительность итерации и сложность задач) не являются достаточными для отображения реальной динамики проектирования из-за неравномерности распределения работ между участниками проекта, выполняющими различные роли.

Решением этой проблемы может быть усовершенствование диаграммы с помощью расширения входных параметров и подходов для увеличения точности отображаемых результатов. Целесообразно разработать адаптивный метод ведения диаграммы, учитывающий показатели скорости команды на каждой фазе разработки. Индикатор такого метода должен отображать выполненный объем задач в каждой контрольной точке, а также разницу между текущей и требуемой скоростью команды.

*Целью* данного исследования является разработка и программная реализация метода расчета и адаптивного прогнозирования оптимального количества ресурсов ИТ проекта по методологии Agile. При разработке метода должны быть учтены как стандартные показатели, характеризующие человеческие ресурсы (рабочие часы, дни, длительность итерации, сверхурочные часы, степень участия того или иного члена команды в проекте, максимальная скорость команды), так и дополнительные показатели, позволяющие повысить наглядность

представления результатов (фокус-фактор, количество незапланированного объема работы, количество оставшейся работы по неделям, разница оценки задач). Целевой аудиторией предлагаемого подхода являются участники Agile команд, скрам-мастер, менеджер проекта, менеджер департамента, заказчик.

## 2. Общие принципы построения диаграммы выполнения задач

Гибкие методологии создания программного обеспечения ориентированы на использование итеративной разработки, динамического формирования требований и обеспечения их реализации в результате постоянного взаимодействия внутри самоорганизующихся рабочих групп, состоящих из специалистов различного профиля. Существует несколько методик, относящихся к классу таких гибких методологий, в частности экстремальное программирование, DSDM (Dynamic Systems Development Method), Scrum, FDD (Feature driven development). Эти методики основаны, как правило, на минимизации рисков путем сведения разработки к серии коротких циклов, называемых итерациями (длительность таких циклов часто принимается двухнедельной). Каждая итерация сама по себе представляет программный проект в миниатюре и включает все задачи, необходимые для выдачи мини-прироста по функциональности: планирование, анализ требований, проектирование, программирование, тестирование и документирование. Хотя отдельная итерация, как правило, недостаточна для выпуска новой версии продукта, подразумевается, что гибкий программный проект готов к выпуску в конце каждой итерации. По окончании каждой итерации команда выполняет переоценку приоритетов разработки. Диаграмма выполнения задач является самым распространенным и эффективным инструментом методологий Agile, отображающим динамику проектирования и используемым для контроля прогресса на любой итерации спринта [1]. На рис. 1 приведена диаграмма, отражающая классический график процесса разработки по методологии Agile.

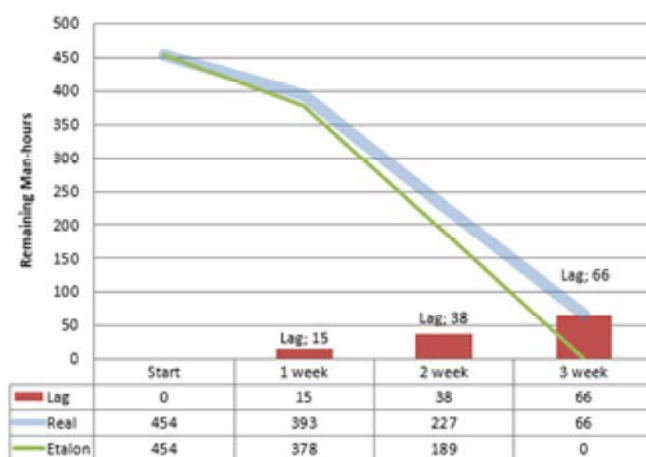


Рис. 1. Классический график процесса разработки

Основные составляющие классического графика процесса разработки: ось X отображает календарные дни итерации; ось Y – объем работы (может измеряться в пользовательских историях, человеко-часах, днях, количестве задач); зеленая кривая отображает ожидаемый прогресс; синяя кривая отображает реальный прогресс и количество оставшейся работы. Зеленая кривая выходит из точки, соответствующей началу итерации и исходному объему работ, а заканчивается на дате завершения итерации. Такой график выполнения задач командой является идеальным и показывает, сколько часов должно быть потрачено ежедневно для успешного выполнения всех задач к окончанию итерации. Реальный график выполнения задач отображается синей кривой, показывающей оставшуюся трудоемкость на каждый день итерации. Если кривая расположена под зеленой линией, то это означает, что команда движется с достаточной скоростью и успевает завершить задачи по итерации к намеченному сроку. Если синяя кривая располагается над зеленой линией, нужно либо повлиять на скорость команды, либо исключить из итерации какие-то задачи, поскольку иначе команда не успеет завершить работу в срок [3]. С помощью графика процесса разработки можно определить не только скорость команды, но и узкие места, препятству-

ющие достижению цели в поставленные сроки. Современные системы отслеживания ошибок поддерживают автоматическую генерацию диаграммы прогресса разработки. Jira – коммерческая система отслеживания ошибок, предназначенная для организации взаимодействия с пользователями. Основной элемент учета в этой системе – задача (ticket). Она содержит название проекта, тему, тип, приоритет, компоненты и содержание. Набор задач формирует базу данных для построения графика. Система Rational Team Concert (RTC) предоставляет график прогресса разработки и управляет жизненным циклом ПО, обеспечивая контекстную коллективную работу для распределенных команд в реальном времени. Система RTC упрощает планирование и выполнение гибких или формальных проектов на основе инструментов планирования и шаблонов. Согласованные процессы, основанные на применении этих систем, помогают повысить качество разрабатываемого программного обеспечения, однако не учитывают в должной мере все существенные различные факторы влияния на график. Кроме того, диаграмма показывает общую скорость команды, однако часто команды состоят из аналитиков, разработчиков, менеджеров и тестировщиков, неравномерно участвующих в процессе. В этом случае непонятно, в какой части процесса возникают трудности. Предлагаемый ниже метод позволяет усовершенствовать диаграммы с помощью расширения входных параметров для увеличения точности отображаемых результатов.

### 3. Разработка метода построения диаграммы выполнения задач

Диаграмма выполнения задач должна содержать набор параметров, которые определяют тренд для построения графика. Такая диаграмма состоит из множества характеристик, используемых в методологиях Agile. Эталонный тренд, представленный на рис. 2, учитывает скорость команды на каждой календарной неделе и распределяет ориентир соответственно [4]. Кривая эталона показывает объем работы, который команда должна выполнить в каждой контрольной точке, чтобы успеть завершить все задачи вовремя.

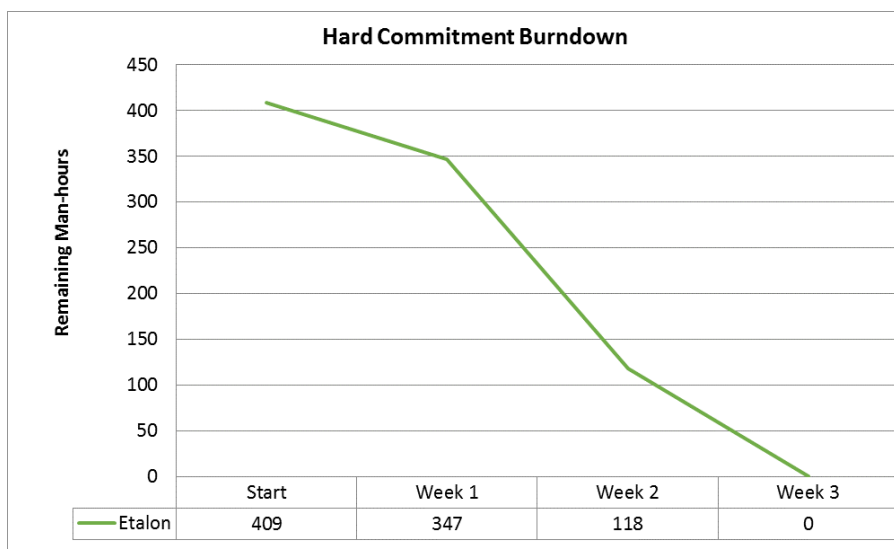


Рис. 2. Эталонный ориентир

В первую и последнюю календарную недели спринта ожидаемый объем выполненных работ намного ниже, чем во вторую, так как начинается и заканчивается спринт в середине недели. Показатели эталонного ориентира оцениваются в человеко-часах – единица учета рабочего времени, которая соответствует часу работы одного человека [5]. Данная единица позволяет оценить работу при планировании более точно, сопоставляя количество работников и сроки выполнения задания. Эталонный ориентир определяется по формуле:

$$E = \{E_{start}, E_{week1}, \dots, E_{weekN}\}, \quad (1)$$

где  $E_{start}$  – предполагаемый объем работы в начале итерации;  $E_{weekN}$  – объем работы в каждой контрольной точке;

$$E_{start} = S + AVG\Delta_{start}, \quad (2)$$

здесь  $S$  – объем работы в итерации;  $AVG\Delta_{start}$  – объем незапланированных задач в начале итерации;

$$E_{weekN} = \frac{E_{weekN-1} - E_{start} \times C_{weekN}}{C_{sprint}}, \quad (3)$$

$C_{weekN}$  – скорость команды в течение недели;  $C_{sprint}$  – скорость команды в итерации.

Каждая задача, планируемая к выполнению за определенный промежуток времени (итерацию), оценивается в идеальных часах. Данный параметр позволяет учитывать все процессы для точного расчета ресурсоемкости проекта на этапе планирования итерации. Время по задачам суммируется, образуя метрику Team Capacity (емкость, объем работы):

$$C_{period} = 8 \times WD * EN - V - SN + O, \quad (4)$$

где  $WD$  – количество рабочих дней;  $EN$  – количество участников проекта;  $V$  – количество дней отпуска;  $SN$  – количество дней, отведенных на больничный;  $O$  – количество часов сверхурочной работы.

При вычислении объема работ учитываются такие показатели, как количество рабочих дней, количество инженеров, отпуска, больничные, сверхурочные часы [6]. Таким образом, команде предоставляется точный индикатор того, с какой скоростью должны быть выполнены задачи.

Второй тренд Score отражает фактическое выполнение задач. Данный показатель, представленный на рис. 3, показывает как команда справляется с задачами.

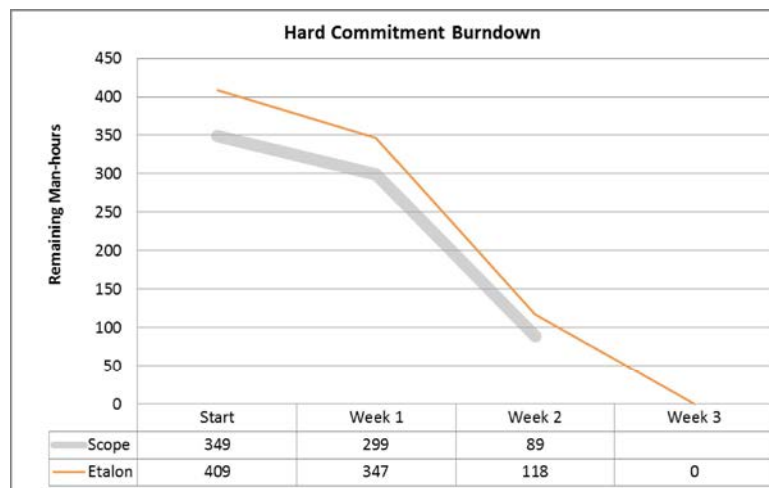


Рис. 3. Выполнение задач на протяжении итерации

В каждой задаче указано, сколько времени уже потрачено на ее реализацию (spent) и сколько еще осталось потратить (remaining). Сумма всех remaining-показателей образует серую линию и определяется по формуле:

$$S = \sum RV, \quad (5)$$

где  $RV$  – количество оставшихся человеко-часов.



Рис. 3. Выполнение задач на протяжении итерации

В классическом виде такая модель недостаточно показательна. Если команда будет руководствоваться только текущим состоянием командной доски, производительность команды не будет отображена в полной мере, так как командная доска, содержащая задачи для текущей итерации, не отражает ряд незапланированных активностей, которые должны быть выполнены (например, исправление дефектов по комментариям от инженера по контролю качества кода и сопутствующее регрессионное тестирование) [7]. Исполнитель имеет также право отклониться от оценок, определенных им ранее. При этом объем запланированных работ к концу итерации всегда будет больше изначального набора задач.

Если команда не будет учитывать прогноз объема незапланированных задач, представленных на рис. 4, то это может привести к возникновению рисков потери качества продукта.

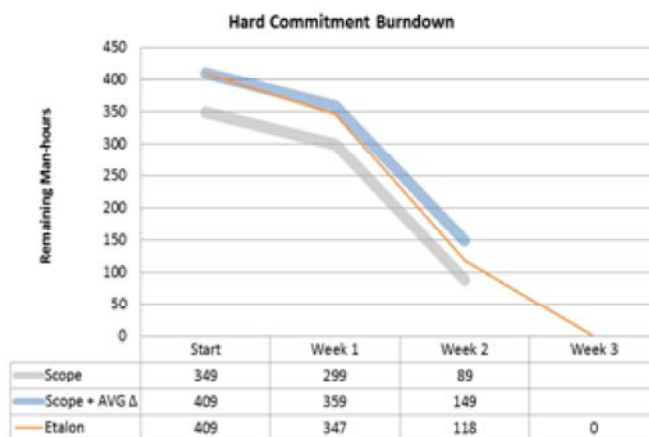


Рис. 4. Нормализованный объем незапланированных задач

Синяя кривая – это показатель нормализованного объема оставшейся работы, в состав которого включен прогноз на незапланированные задачи и превышение оценок. Этот показатель определяется следующим образом:

$$S + AVG\Delta = S * UTC * EDC, \quad (6)$$

где **UTC** – количество незапланированных задач; **EDC** – мера превышения оценок.

Прогноз незапланированных задач и превышения оценок осуществляется для расчета наиболее точного ожидаемого результата в конце итерации. Современные системы отслеживания ошибок не учитывают данные показатели, которые значительно могут повлиять на работу команды. Если прогноз на незапланированный объем задач не будет учтен, это может способствовать возникновению рисков, которые повлекут за собой: снижение качества раз-

рабатываемого продукта; увеличение ресурсов, рабочих часов, что в свою очередь повлияет на финансовую составляющую проекта; расторжение контракта с заказчиком; несоблюдение Agile методологии.

Прогнозируемые характеристики определяются как взвешенное среднее результатов предыдущих спринтов. Вес конкретной итерации определяется консолидированной оценкой, руководствующейся набором адаптивных правил. В показатель включается вес всех незапланированных задач. Прогноз незапланированного объема задач определяется по формуле:

$$US_{forecast} = \frac{S \times (US_{AVG} - US_{current})}{100\%}, \quad (7)$$

где  $US_{AVG}$  – взвешенное среднее результатов предыдущих спринтов;  $US_{current}$  – количество незапланированных задач в текущей итерации.

Объем незапланированных задач для текущей итерации определяется следующим образом:

$$US_i = \frac{\sum RV_i}{S + AVG \Delta_i \times 100\%}, \quad (8)$$

здесь  $\sum RV_i$  – количество незапланированных задач в текущей итерации.

Взвешенное среднее результатов предыдущих спринтов определяется по формуле:

$$US_{AVG} = \frac{\sum US_i \times USW_i}{\sum USW_i}, \quad (9)$$

где  $USW_i$  – коэффициент незапланированных задач.

Вторым показателем, влияющим на прогноз, является отклонение от ошибки. Каждая задача на командной доске содержит изначальную оценку, которая была проставлена исполнителем для выполнения данной задачи, и фактически потраченное время.

Сумма разниц этих показателей отображает превышение изначальной суммарной оценки. Процедура корректировки коэффициента влияния определяется для конкретной итерации индивидуально по таким критериям: нахождение наиболее аномальных превышений оценки; выявление причин превышения; определение превентивных мер.

Прогноз превышения оценок находится по формуле:

$$ED_{forecast} = \frac{(S + US_{forecast}) \times ED_{AVG}}{100\%}, \quad (10)$$

где  $ED_{AVG}$  – взвешенное среднее результатов предыдущих спринтов.

Погрешность в оценке в текущей итерации вычисляется по формуле:

$$ED_i = \left( \frac{\sum SV + RV - OV}{S_i} \right) \times 100\%, \quad (11)$$

здесь  $SV$  – потраченное время на выполнение задачи;  $RV$  – оставшееся время;  $OV$  – запланированное время.

Отклонение от оценки в текущей итерации определяется по формуле:

$$ED_{AVG} = \frac{\sum ED_i \times EDW_i}{\sum EDW_i}, \quad (12)$$

где  $EDW_i$  – коэффициент превышения оценок.

По результатам построения эталонной и действительной тенденций рассчитываются основные показатели эффективности команды [8]. Первый показатель демонстрирует разницу между запланированным и выполненным объемом, т.е. насколько команда отстает от графика или его опережает (с учетом прогноза на отклонение). Пример применения показателя эффективности команды представлен на рис. 5.

Данный параметр (в человеко-часах) определяется следующей суммой:

$$LAG = S + AVG \Delta - E. \quad (13)$$

Если команда выполняет задачи равномерно, следуя плану, то отклонение от графика будет равняться нулю. В случае, когда параметр Lag отрицательный, то это сигнализирует

о том, что команда опережает план. Если же он положительный, это может свидетельствовать о том, что команда не успевает выполнить работу в поставленный срок. Данную ситуацию можно объяснить следующими причинами: увеличение сложности одной или нескольких пользовательских историй; сокращение ресурсов; ошибка в изначальной оценке на этапе планирования итерации; нахождение большого количества дефектов после фазы тестирования; увеличение объема смежного функционала, который был задействован в момент разработки; недостаток опыта и знаний в команде для решения поставленных задач.

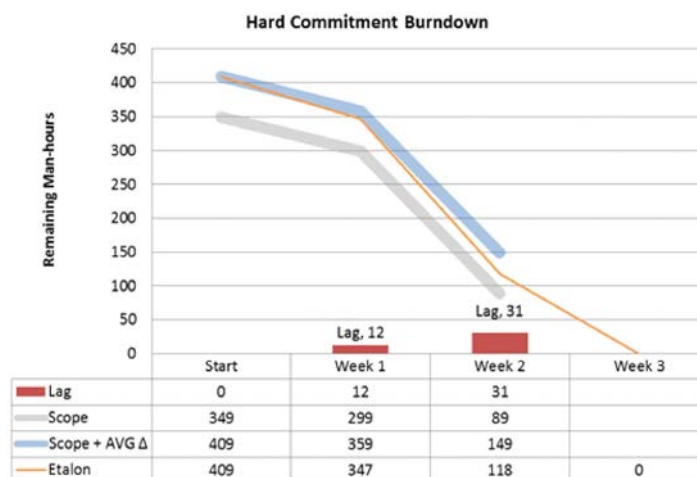


Рис. 5. Эффективность с учетом прогноза на отклонение

Данный показатель необходим команде для устранения рисков на этапе их формирования. Причины возникновения рисков могут быть выявлены с помощью таких показателей как фокус-фактор и количество человеко-часов в текущей итерации. Описанная выше модель позволяет команде контролировать выполнение задач в итерации, но не отображает эффективность расхода командой временных ресурсов. Фокус-фактор (focus factor) – это индикатор, показывающий, какую часть времени команда тратит на непосредственное выполнение запланированных задач. Данный показатель может быть смещен различными шумами (например, неспособностью или нежеланием команды фиксировать потраченное время на каждом этапе разработки). Фокус-фактор рассчитывается по формуле:

$$FF = \frac{WD}{C} \times 100\%, \quad (14)$$

где **WD** – количество человеко-часов, которые были потрачены на реализацию; **C** – скорость команды.

Нижняя допустимая граница командного фокус-фактора в Agile методологиях принимается равной 75%. Это означает, что команда должна тратить более 3/4 рабочего времени на выполнение задач на командной доске для успешной реализации пользовательской истории. Графическое отображение примера командного фокус-фактора представлено на рис. 6.

Чтобы наиболее точно идентифицировать причину низкого командного фокус-фактора, следует учитывать данный параметр для разработчиков, тестировщиков и других членов команды отдельно. Например, общий показатель может быть меньше 75% по причине недостаточной эффективности работы бизнес-аналитиков. В таком случае менеджер точно определит проблему, вызывающую отставание, и устранит ее. Дополнительным индикатором, отображающим эффективность команды, является скорость выполнения пользовательских историй.

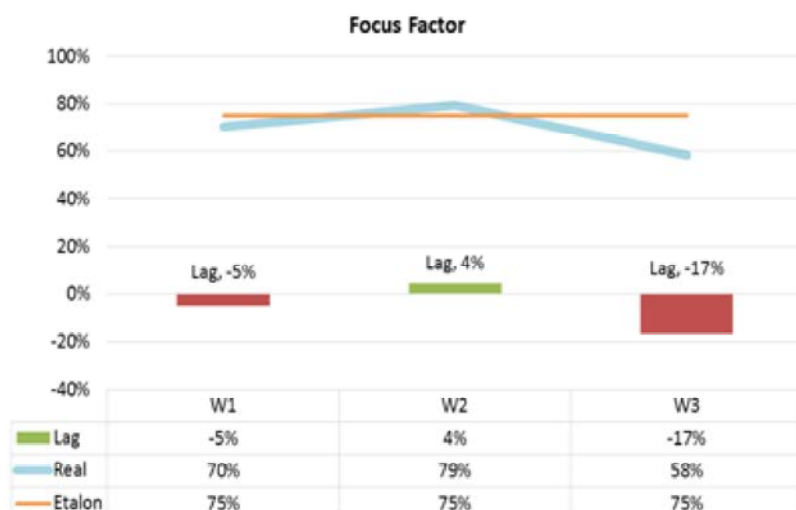


Рис. 6. Пример командного фокус-фактора

Для наглядности данный параметр рассматривается с учетом максимальной, эталонной и действительной скорости команды. Максимальным ресурсам, на которые может рассчитывать команда, соответствует полное привлечение сторонних разработчиков, тестировщиков и других участники проекта. Эталонный показатель учитывает лишь текущие ресурсы, без учета неполной занятости или отсутствия некоторых участников команды. Действительная скорость – скорость в текущей итерации, в которой были учтены больничные, отпуска, сверхурочные часы.

#### 4. Программная реализация

Для реализации поставленной задачи было разработано Web-приложение на основе схемы использования шаблонов MVC, позволяющее разделять данные, представление и обработку действий пользователя на три отдельных компонента: модель, представление, вид. Для создания более гибкой архитектуры был использован Spring Framework, обеспечивающий решение многих задач для создания информационной системы с применением языка Java. При работе с приложением пользователю предоставляется возможность использовать различные формы для заполнения данных: количество участников команды, роль и занятость, количество рабочих дней, даты контрольных точек. Пример создания новой итерации представлен на рис.7.

Рис. 7. Форма создания спринта

К созданной итерации пользователь может добавить список проектных задач, содержащий характеристики каждой записи. Например, страница «Sprint Info», представленная на рис. 8, содержит информацию о текущей итерации: количество участников и их роли в

итерации, расчет текущих ресурсов, скорость команды, объем запланированной и выполненной работы, фокус-фактор, прогноз на незапланированный объем задач и погрешность в оценке, график выполнения задач. Также на странице представлен прогноз объема незапланированных задач и превышения оценок. Результаты прогноза отображаются, начиная с третьей итерации, когда система хранит достаточное количество данных для прогнозирования. С каждой итерацией оценка показателей эффективности становится точнее. Данные показатели можно проанализировать в любой фазе итерации, что делает модель гибкой и легко масштабируемой.



Рис. 8. Объем выполненных задач на протяжении итерации

На странице «Graphics» представляется диаграмма выполнения задач, в построении которой участвуют все атрибуты и параметры, описанные выше (рис. 9).



Рис. 9. График выполнения задач

Ось Y отображает количество человеко-часов, ось X – контрольные точки, которые были определены при создании итерации согласно рабочим неделям. Первая контрольная точка – первый день итерации, четвертая – последний. Пользователю предоставлена возможность корректировать даты контроля для мониторинга эффективности на любой фазе спринта. Диаграмма выполнения задач состоит из трех кривых и показателя Lag. Серая кривая на графике отображает фактическое выполнение задач. Данный показатель отображает, с какой скоростью команда выполняет проектные задачи согласно доске. В каждой задаче указано, сколько времени уже потрачено на ее реализацию и сколько осталось потратить. Сумма всех человеко-часов, необходимых для выполнения задачи, соответствует серой кривой. Эталонный ориентир представляется зеленой кривой, которая показывает, какой объем работ команда должна завершить в каждой

контрольной точке, чтобы успеть выполнить все задачи вовремя. Эталонный тренд учитывает скорость команды на каждой календарной неделе и распределяет ориентир соответственно. Синяя кривая соответствует нормализованному объему оставшейся работы, в состав которого включен прогноз на незапланированные задачи и превышение оценок. Последний показатель Lag, представленный на рис. 9, наглядно демонстрирует разницу между эталонной и действительной тенденцией, т.е. насколько команда отстает или же опережает график с учетом прогноза на отклонение. Пользователю предоставляется возможность сохранить график в различных форматах: PNG, JPEG, PDF, SVG. Таким образом, документ может быть распечатан так, как он отражается на экране пользователя. Кроме того, генерация PDF файлов является удобным способом документирования. Разработанный программный продукт характеризуется открытостью, кроссплатформенностью и распространенностью, обеспечивая точность и неизменность передачи данных по цепочке «генерация - просмотр – печать».

## 5. Выводы и перспективы дальнейших исследований

Современные системы отслеживания ошибок основаны на различных способах визуализации хода разработки программного обеспечения для IT проектов. Управление проектами представляет собой последовательный, поступательный способ координации деятельности в сфере разработки программного обеспечения и информационных технологий. Основой построения плана команды в гибких методологиях разработки Agile является динамическое формирование диаграммы выполнения задач.

Предложенный метод позволяет усовершенствовать диаграммы с помощью расширения входных параметров для увеличения точности отображаемых результатов. Прогнозируемые характеристики определяются при этом как взвешенное среднее результатов предыдущих спринтов. Вес конкретной итерации устанавливается консолидированной оценкой, руководствующейся набором адаптивных правил. В показатель включается вес всех незапланированных задач.

Разработанное для реализации предложенного метода Web-приложение на основе схемы использования шаблонов MVC позволяет осуществлять представление и обработку действий пользователя по трем отдельным компонентам (модель, представление, вид). Для создания более гибкой архитектуры был использован Spring Framework, обеспечивающий решение многих задач для создания информационной системы с применением языка Java.

Перспективным развитием рассмотренного подхода является расширение его функциональности на основе комплексного применения наиболее перспективных методологий управления IT проектами, что позволит избавиться от привязки к конкретным видам разработки.

**Список литературы:** 1. *Cohn, M. Succeeding with Agile: Software Development Using Scrum* [Текст] / Cohn, M. W.: Pearson Education, 2009. 114 p. 2. *Martin R. C. Agile Software Development, Principles, Patterns, and Practices* [Текст] / Martin, R. C. N.: Pearson Education, 2002. 458 p. 3. *Rubin K. S. Essential Scrum: A Practical Guide to the Most Popular Agile Process* [Текст] / Rubin K. S. H.: Addison-Wesley Professional, 2012. 281 p. 4. *Maurya, A. Running Lean: Iterate from Plan A to a Plan That Works* [Текст] / Maurya, A. W.: O'Reilly Media, 2012. 129 p. 5. *Ries E. The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses* [Текст] / Ries, E. W.: Crown Business, 2011. 90 p. 6. *Croll A. Lean Analytics: Use Data to Build a Better Startup Faster* [Текст] / Croll, A. R.: O'Reilly Media, 2013. 370 p. 7. *Rasmusson J. Agile Samurai: How Agile Masters Deliver Great Software* [Текст] / Rasmusson, J. W.: Pragmatic Bookshelf, 2010. 237 p. 8. *Fowler M. Refactoring: Improving the Design of Existing Code* [Текст] / Fowler, M. L.: Addison-Wesley Professional, 1999. 281 p.

*Поступила в редколлегию 11.12.2015*

**Жмаева Юлия Владимировна**, студентка каф. Искусственного интеллекта ХНУРЭ Адрес: Украина, 61166, Харьков, пр. Науки,14.

**Удовенко Сергей Григорьевич**, д-р техн. наук, профессор кафедры электронных вычислительных машин ХНУРЭ. Адрес: Украина, 61166, Харьков, пр. Науки,14, тел. 057-7021453, e-mail: serhii.udovenko@nure.ua.

**Чалая Лариса Эрнестовна**, канд. техн. наук, доцент кафедры искусственного интеллекта ХНУРЭ. Адрес: Украина, 61166, Харьков, пр. Науки, 14, тел. 057-7021337, e-mail: larysa.chala@nure.ua.

## **МЕТОД НЕЙРОСЕТЕВОЙ ОБРАБОТКИ ЗАШУМЛЕННЫХ ПРОСТРАНСТВЕННЫХ ДАННЫХ**

---

Предлагается метод нейросетевой обработки зашумленных картографических данных геоинформационной системы. Метод предполагает реализацию на параллельных вычислительных структурах процедур предварительной фильтрации полутоновых пространственных изображений и помехоустойчивого детектирования контуров объектов изображений. Осуществляется программная реализация метода и приводятся результаты его тестового моделирования.

### **1. Введение и постановка задачи**

Для представления объектов в геоинформационных системах (ГИС) используют пространственные и атрибутивные типы данных [1]. Под пространственными данными в общем случае понимают сведения, которые характеризуют местоположение объектов в пространстве относительно друг друга и их геометрию.

Задачей обработки пространственных данных в ГИС может быть как улучшение (восстановление) изображения по некоторому критерию, так и специальное преобразование, целенаправленно меняющее изображение. В последнем случае обработка изображений может быть промежуточным этапом для дальнейшего распознавания изображения (например, для выделения контура объекта). Методы обработки изображения могут существенно различаться в зависимости от того, каким путем изображение было получено (синтезировано системой машинной графики, путем оцифровки черно-белой или цветной фотографии или видеокadra). Если изображения были получены с помощью оцифровки, то они, как правило, искажены шумами различной природы. Источники шума могут быть различными: неидеальное оборудование для захвата изображения камерой, погрешности камеры или сканера, плохие условия съемки (например, шумы, возникающие при ночной фото/видеосъемке), помехи при передаче по аналоговым каналам (наводки от источников электромагнитных полей, собственные шумы активных компонентов линии передачи) неточность выделения яркостного и цветоразностных сигналов из аналогового композитного сигнала [2]. Шумы бывают разных типов, в частности: белый шум – сигнал, отсчеты которого не коррелируют друг с другом, и его разновидность – белый гауссовский шум, возникающий, в частности, при плохих условиях приема сигнала; импульсный шум – случайные изолированные точки на изображении, значение которых сильно отличается от значений окружающих их точек (обычно возникает при передаче изображений по аналоговому каналу).

Шумоподавление в задачах обработки пространственных данных в ГИС служит для улучшения визуального восприятия анализируемых изображений, но может также использоваться для специализированных целей, например, для увеличения четкости при выделении контуров объектов, для предобработки и последующего распознавания и т.п. Также шумоподавление играет важную роль при сжатии изображений. Одной из основных проблем в алгоритмах сжатия является определение локальной зашумленности анализируемой области изображения, поскольку при сжатии сильный шум может быть принят за детали изображения, что может привести к увеличению сложности сжатия и отрицательно повлиять на результирующее качество сжатого изображения. Наиболее адекватными с точки зрения использования в практических задачах являются модели аддитивного Гауссова и импульсного шумов. Аддитивный Гауссов шум характеризуется добавлением к каждому пикселю изображения значений из соответствующего нормального распределения с нулевым средним значением.

Выделение границ (контуров) занимает особое место в перечне задач обработки изображений в ГИС, связанных с последующим поиском и выделением объектов. Оно основывается на алгоритмах, которые детектируют точки цифрового изображения, в которых

резко изменяется яркость или присутствуют другие виды неоднородностей. Алгоритмы выделения границ и привязки их к существенным элементам изображения называют детекторами границ или детекторами краев (edge detector) [3]. Результатом выделения границ является набор связанных кривых, обозначающих границы объектов, граней и отрисовок на поверхности, а также кривых, которые отображают изменения положения поверхностей. Таким образом, применение фильтра выделения границ к изображению может существенно уменьшить количество обрабатываемых данных, так как отфильтрованная часть изображения считается менее значимой, а наиболее важные структурные свойства изображения сохраняются. Детекторы границ используют относительно локальные фрагменты изображения, на основании анализа которых делается вывод о наличии краевых точек – точек изображения, в которых яркость изменяется особенно сильно. Совокупность краевых точек формирует края областей изображения. Как правило, детекторы границ дают удовлетворительный результат при наличии резких локальных изменений яркости. Однако в условиях наличия шумов и помех существенным образом искажается информация о положении краевых точек, что приводит к появлению двух типов возможных ошибок. Первый тип связан с пропуском реально существующей краевой точки, а второй – с появлением ложной. Улучшение качества функционирования детекторов границ требует увеличения площади фрагмента изображения, используемого для принятия решения о наличии краевой точки, а также сложных алгоритмов предварительной обработки. С точки зрения реализации это означает необходимость выполнения значительного объема вычислений. Одним из возможных вариантов реализации задач обработки зашумленных пространственных данных, в частности, помехоустойчивого детектирования границ, является использование нейросетевых методов, позволяющих использовать параллельные схемы попиксельной обработки изображений с помощью искусственных нейронных сетей (ИНС) [4].

*Целью данного исследования* является разработка и программная реализация метода нейросетевой обработки цифровых полутоновых пространственных изображений, позволяющего повысить качество и помехоустойчивость выделения контуров объектов ГИС.

## **2. Общие принципы шумоподавления в задачах обработки цифровых изображений**

Алгоритмы шумоподавления (фильтрации шумов) обычно специализируются на подавлении какого-то конкретного вида шума. Многие шумы можно достаточно точно описать моделью белого Гауссова шума, поэтому большинство алгоритмов фильтрации ориентировано на его подавление. К наиболее распространенным методам фильтрации шумов относятся сглаживающие фильтры, фильтры Винера, медианные фильтры и ранжирующие фильтры. Для подавления Гауссова шума используются как линейные, так и нелинейные фильтры. Линейный фильтр определяется вещественнозначной функцией (ядром фильтра), заданной на растре. При линейной сглаживающей фильтрации значение интенсивности в каждой точке усредняется по некоторой сглаживающей маске. При осуществлении фильтрации яркостные характеристики каждой точки цифрового изображения заменяются другим значением яркости, которое признается в наименьшей степени искаженным помехой. Выделяют частотную и пространственную фильтрацию [5]. Частотные методы преобразований изображений основываются на применении преобразования Фурье. Данный подход позволяет обрабатывать функцию в частотной области, после чего без потери информации вернуться к исходному виду. Пространственные методы улучшения изображений, как правило, применяются к растровым изображениям, представленным в виде двумерных матриц. Принцип пространственных алгоритмов заключается в применении специальных оконных операторов (масок) к каждой точке исходного изображения. Чаще всего маска представляет собой небольшой двумерный массив, а методы улучшения, базирующиеся на таком подходе, называют фильтрацией по маске. При осуществлении линейной фильтрации отклик маски задается суммой произведений пикселей в области покрытия фильтра. В качестве линейного сглаживающего фильтра используется усредняющий фильтр, выходным значением которого является среднее значение по окрестности маски фильтра. Операции, выполняемые нелинейным пространственным фильтром, зависят от значений элементов матрицы изображения, которые находятся в анализируемой окрестности. При

медианной фильтрации значения пикселей представляют собой усредненное значение точек соответствующей окрестности. Часто при устранении шума медианный фильтр является более эффективным, чем обычное усреднение, так как приводит к меньшим искажениям границ выделяемых объектов. В качестве маски при медианной фильтрации используется двумерное окно с центральной симметрией, при этом его центр располагается в текущей точке фильтрации. В основу адаптивной фильтрации положен винеровский фильтр, являющийся одним из типов линейного фильтра для адаптивной локальной обработки изображений. Преимущество адаптивного фильтра состоит еще и в том, что он сохраняет края и другие фрагменты объектов изображения. В то же время винеровский фильтр требует большего времени для вычислений, чем линейный фильтр. Среднее значение яркости вычисляется для центрального пикселя маски, содержащей значения яркости исходного изображения в покрытой маской локальной области изображения.

Простейшая идея удаления шума – усреднять значения пикселей в пространственной окрестности. Для каждого пикселя анализируются соседние для него пиксели, которые располагаются в прямоугольном окне вокруг этого пикселя. Медианная фильтрация – стандартный способ подавления импульсного шума. Для каждого пикселя в некоторой его окрестности (окне) определяется медианное значение. Однако в чистом виде медианный фильтр размывает мелкие детали, величина которых меньше размера окна для поиска медианы.

Метод главных компонент (МГК) позволяет выделить структуру в многомерном массиве данных. Он применяется в основном для распознавания или для сжатия изображений с белым гауссовским шумом и основан на нахождении таких базисных векторов исследуемого пространства, которые бы наилучшим образом отражали расположение исходных данных в этом пространстве. При этом все изображение разбивается вначале на блоки, которые обрабатываются независимо и должны располагаться с небольшим перекрытием, чтобы избежать артефактов блочности при их стыковке. МГК хорошо работает на текстурованных областях (после подавления шума текстура останется по-прежнему четкой).

Конечный результат анализа изображений во многом определяется качеством сегментации, а степень детализации выделяемых характеристик зависит от конкретной задачи. В настоящее время не существует универсального метода или алгоритма для решения всех типов задач сегментации. В большинстве случаев выбирается один или несколько алгоритмов, и модифицируются под специфичные условия задачи. Сегментация решает в общем смысле две основные задачи: разделение изображения на части в целях осуществления дальнейшего локального анализа; изменение формы описания элементов изображения, что позволяет представить точки как высокоуровневые структуры, обеспечивающие эффективность дальнейшего анализа изображения.

Принцип работы методов сегментации цифровых изображений основан на разности яркости элементов и фона изображения [6]. Математическое обоснование этих методов заключается в вычислении производных, представленных для цифровых изображений в виде дискретных приближений градиента. В качестве градиентных методов выделяют операторы, представляющие собой матричные маски: перекрестный оператор Робертса, оператор Превитта, оператор Собеля и оператор Канни. Рассмотрим некоторые особенности этих операторов. Оператор Канни позволяет, как правило, обеспечить: хорошее обнаружение (повышение отношения сигнал/шум); хорошую локализацию (правильное определение положения границы); единственный отклик на одну границу. Для уменьшения чувствительности алгоритмов сегментации к шуму применяется фильтрация по первой производной, несколько размывающая исходное изображение (после вычисления градиента сглаженного изображения в контуре границы оставляются только точки максимума градиента изображения). Далее используется информация о направлении границы для того, чтобы удалять точки именно рядом с границей и не разрывать саму границу вблизи локальных максимумов градиента. Для определения направления градиента можно использовать оператор Собеля, округляющий полученные значения направлений до одного из четырех углов (0, 45, 90 и 135 градусов) и удаляющий с помощью порогов слабые границы.

Для решения задачи выделения перепадов яркости можно применить дифференциальные операторы более высокого порядка, например оператор Лапласа:

$$\Delta^2 f(x_1, x_2) = \frac{\delta^2 f(x_1, x_2)}{\delta x_1^2} + \frac{\delta^2 f(x_1, x_2)}{\delta x_2^2}. \quad (1)$$

Данный оператор обнаруживает границы в местах смены знака производной функции яркости. Роль лапласиана в задачах сегментации сводится к использованию его свойства пересечения нулевого уровня для локализации контура и выяснению, на темной или на светлой стороне контура находится рассматриваемый пиксель  $(x; y)$ . Основным недостатком лапласиана является высокая чувствительность к шумам и возможность появления разрывов в контуре. К его достоинствам можно отнести то, что он нечувствителен к ориентации границ областей и имеет малую ресурсоемкость. Для уменьшения влияния шумов часто используют лапласиан в сочетании со сглаживанием по методу Гаусса. Такое сочетание называют оператором лапласиан гауссиана (Laplacian of Gaussian – LoG). К его достоинствам можно отнести слабую чувствительность к шумам и ориентации границ областей, что позволяет четко выделять внешний и внутренние контуры объекта. Для вычисления дискретного градиента можно применять оператор Робертса. Поскольку для вычисления модуля градиента можно использовать разности в любых двух взаимно перпендикулярных направлениях, в операторе Робертса берутся диагональные разности. К недостаткам этого оператора относятся высокая чувствительность к шумам и ориентации границ областей, а также возможность появления разрывов в контуре и отсутствие явно выраженного центрального элемента. Оператор Робертса прост в реализации и обладает высоким быстродействием, однако сильно чувствителен к помехам. На практике для вычисления дискретных градиентов чаще всего используются операторы Превитта и Собеля, дающие нулевой отклик на областях постоянной яркости. При наличии центрального элемента и малой ресурсоемкости этим операторам свойственна высокая чувствительность к шумам и ориентации границ областей, а также возможность появления разрывов в контуре. Анализ методов фильтрации показывает, что для импульсных помех наиболее подходит медианный фильтр, хорошо сохраняющий границы элементов и обладающий высоким быстродействием. Градиентные алгоритмы выделения контуров элементов изображений просты в программной реализации, однако результаты, получаемые с их помощью, зависят от качества исследуемых изображений.

### 3. Метод нейросетевой обработки зашумленного изображения

Предлагаемый метод предполагает реализацию на параллельных вычислительных структурах процедур предварительной нейросетевой обработки зашумленных изображений (процедура 1) и последующего выделения границ (процедура 2).

**Процедура 1.** Функционирование ИНС, применяемой для улучшения визуального качества цифровых изображений, оценивается по последовательности ее выходных сигналов. Для предварительной обработки исходных изображений можно использовать мультипликативную подстройку их яркости. ИНС обрабатывает пиксели с оцениванием локальных характеристик, поэтому для более эффективной обработки представляется целесообразным введение дополнительного этапа автоматической настройки уровней яркости, реализованной в стандартных графических пакетах. Предлагаемый трехэтапный способ предварительной нейросетевой обработки изображений содержит следующие этапы: предобработка яркости исходного изображения; обработка изображения на локальном уровне с использованием ИНС; обработка изображения на глобальном уровне с применением алгоритма самонастройки уровней яркости.

В процедуре 1 используется локально-адаптивный подход к обработке изображений. Особенность этого подхода заключается в независимой обработке каждого пикселя изображения, исходя из имеющегося набора его локальных и глобальных характеристик (например, локальная и глобальная средняя интенсивность). В отличие от известных подходов, где полнопиксельное изображение поступает для обработки на вход ИНС (например, сети Хопфилда и карты Кохонена), в рассматриваемом способе обработки изображений ИНС обучается попиксельной обработке. При этом уменьшаются требования к объему оперативной памяти, необходимой для хранения информации об ИНС. Такой подход допускает

объединение множества ИНС, обрабатывающих по одному пикселю, в двухмерный массив для распараллеливания вычислений.

ИНС используется для аппроксимации преобразования следующего вида:

$$L^*(x, y) = T\left(L(x, y), D_{(x,y)}, m_{(x,y)}\right), \quad (2)$$

где  $L^*(x, y)$  и  $L(x, y)$  – соответственно обработанное и исходное значения яркости пикселя  $(x, y)$ ;  $m_{(x,y)}$  и  $D_{(x,y)}$  – соответственно средние значения яркости и дисперсии яркости в локальной окрестности обрабатываемого пикселя.

Для рассматриваемого преобразования (2) ИНС, аппроксимирующая функцию  $T$ , должна иметь 3 входа и 1 выход. При обработке цветных изображений сначала осуществляется их преобразование в полутоновые, затем производится их обработка с использованием ИНС, а после этого информация о цвете восстанавливается [7].

Для оценки качества обработанных изображений во время ИНС обучения будем использовать оценку качества  $f_{\text{mod}}$ , которая является модификацией оценки Мунтеану-Роса  $f_{\text{MR}}$  и имеет следующий вид:

$$f_{\text{mod}} = \frac{MN - \eta}{MN} + \frac{256 - \exp(H)}{192} + \frac{255 - E / \eta}{255}; \quad H = -\sum_{i=1}^{256} l_i \log l_i, \quad (3)$$

где  $E$  – суммарная интенсивность пикселей на контурах на изображении;  $\eta$  – количество пикселей на контурах;  $N$  и  $M$  – соответственно ширина и высота изображения;  $l_i$  – доля пикселей с  $i$ -м уровнем яркости.

Данная оценка основана на том, что для визуального восприятия большее значение имеет перепад яркости в соседних пикселях, чем значение яркости в каждом пикселе, что позволяет сделать вывод о целесообразности максимизации  $E$  и  $\eta$ . В то же время, присутствие большого числа уровней градации яркости на изображении и равномерность гистограммы изображения, определяющей значение  $H$ , положительно сказывается на восприятии изображения. Уменьшение значения  $f_{\text{mod}}$  соответствует улучшению визуального качества рассматриваемого изображения. Было проведено экспериментальное сравнение оценки  $f_{\text{mod}}$  с оценкой  $f_{\text{VIF}}$  – наиболее известной объективной оценкой качества VIF (Visual Information Fidelity), основанной на сравнении исходного и измененного изображений, которая учитывает количество информации на изображениях и целостность сигнала на измененном изображении по сравнению с исходным. Корреляция между субъективной оценкой качества  $f_{\text{mod}}$  и эталонной оценкой  $f_{\text{VIF}}$  (0,59) существенно выше корреляции между оценками  $f_{\text{MR}}$  и  $f_{\text{VIF}}$  (0,23), что подтверждает целесообразность применения функции (3) для анализа качества процедур обработки изображений. Для повышения скорости обработки изображения на локальном уровне процедуры 1 предложены следующие оценочные формулы для вычисления локальных значений среднего  $m_{(x,y)}$  и дисперсии  $D_{(x,y)}$ :

$$\tilde{m}_{(x,y)} = \frac{m_{(x)}(y_\lambda) + m_{(y)}(x_\kappa) + \langle L \rangle}{3}, \quad (4)$$

$$\tilde{D}_{(x,y)} = \frac{D_{(x)}(y_\lambda) + D_{(y)}(x_\kappa)}{2}, \quad (5)$$

где  $m_{(x)}(y_\lambda)$  и  $m_{(y)}(x_\kappa)$  – средняя яркость в столбце  $y_\lambda$  и строке  $x_\kappa$  соответственно;  $D_{(x)}(y_\lambda)$  и  $D_{(y)}(x_\kappa)$  – дисперсии яркости в столбце  $y_\lambda$  и строке  $x_\kappa$  соответственно;

$y_\lambda \in [i_1; i_2]$ ,  $x_k \in [j_1; j_2]$ ;  $\langle L \rangle$  – средняя яркость изображения. Исследование зависимостей (4) и (5) показало, что для реальных изображений соотношение «сигнал-шум» при использовании составляет около 25 дБ для приближенного среднего и около 32 дБ для приближенной дисперсии. Применение в процедуре 1 оценочных формул позволяет ускорить вычисления в среднем в 20 раз для окрестности размером 65x65 пикселей. Для обработки изображения на глобальном уровне (этап 3 процедуры 1) целесообразно использовать алгоритм самонастройки уровней яркости MSR (Multi-Scale Retinex) [6]. В процедуре 1 предлагается применять модифицированный алгоритм генетической оптимизации (ГАИНС) для определения оптимального варианта структуры ИНС с логсигмоидной функцией активации. Каждая структура кодируется при этом по списку имеющихся связей. В ГАИНС используются оригинальные операторы кроссовера и мутации, учитывающие структуру ИНС. Размер популяции может динамически настраиваться в процессе поиска оптимального решения (с функцией приспособленности  $f_{mod}$ ). Структуры ИНС во время обучения оцениваются по визуальному качеству обработанных изображений. При этом для снижения вычислительных затрат может быть использован упрощенный вариант оценки (3):

$$f = \frac{N * M - \mu}{N * M} + \frac{256 - \exp(H)}{192}, \quad H = - \sum_{i=1}^{256} l_i \log l_i, \quad (6)$$

где  $N$  и  $M$  – ширина и высота изображения соответственно;  $\mu$  – количество пикселей на границах перепадов уровней яркости;  $l_i$  – доля пикселей с  $i$ -м уровнем яркости.

В этом случае в качестве функции приспособленности для алгоритма ГАИНС используется функция (6). Первое слагаемое функции  $f$  необходимо для максимизации числа пикселей на границах перепадов яркости, что улучшает детализацию изображения. Второе слагаемое функции  $f$  предотвращает опасность вырождения обработанного изображения в бинарное.

**Процедура 2.** Принцип нейросетевого выделения контуров на изображении, предварительно улушенном с помощью процедуры 1, представлен на рис.1.

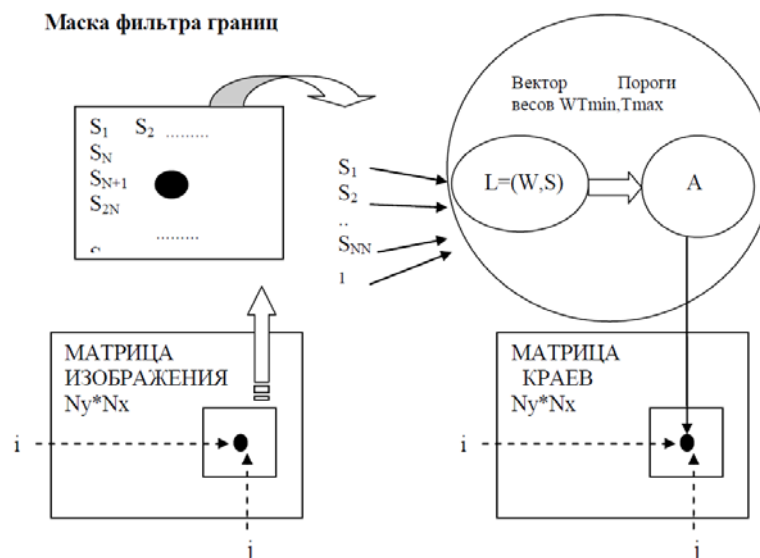


Рис. 1. Обработка изображения нейросетевым детектором границ

Исходное изображение представляет собой матрицу значений яркости размером  $N_y * N_x$ . Результатом обработки этой информации является матрица описания контуров обрабатываемого изображения также размером  $N_y * N_x$ . Для формирования значения каждого элемента  $(i,j)$  матрицы контуров (краев) используются  $N * N$  элементов матрицы исходно-

го изображения (после процедуры 1), формирующих квадратную маску фильтра границ. Таким образом, для принятия решения о принадлежности точки  $(i,j)$  к подмножеству краевых точек применяется  $N*N$  близлежащих элементов. Эти элементы упорядочиваются и из них формируется вектор  $S$  размерности  $N_{sen} = N*N$ , содержащий значения яркости в окрестности  $(i,j)$ . Размерность вектора  $S$  на единицу больше, чем количество используемых точек. Компонента  $N+1$  вектора  $S$  имеет фиксированное значение 1, что обеспечивает возможность учета постоянной составляющей.

Вектор показаний сенсоров  $S$  подается на вход нейросетевого детектора границ (НСДГ), содержащего линейную ( $L$ ) и нелинейную ( $A$ ) части. Линейная часть НСДГ, имеющая  $N+1$  вход, выполняет скалярное умножение вектора  $S$  на  $(N+1)$ -мерный вектор весов  $W$ . Нелинейная часть  $A$  представляет собой кусочно-линейную активационную функцию, описываемую парой  $(Tmin, Tmin)$  и  $(Tmax, Tmax)$ . Эта часть НСДГ может иметь сложную структуру, выполняющую последовательно несколько кусочно-линейных преобразований с заданными параметрами. Обработка изображения однослойным НСДГ состоит из  $K_y * K_x$  одностипных преобразований элементов исходного изображения. Параметрами обработки являются вектор весов  $W$  линейной части и параметры активационной функции нелинейной части. Такая структура фильтра предполагает возможность полного распараллеливания. С практической точки зрения интерес представляет подход, основанный на параллельно-последовательной схеме обработки изображения. В этом случае реализуется структура НСДГ, состоящая из некоторого количества фильтров, параллельно осуществляющих базовое преобразование. Синтез такого детектора подразумевает выбор структуры ИНС, выбор активационных функций и определение значений весовых коэффициентов. В процедуре 2 (детектирование границ изображения) используется многослойный персептрон (МСП) с кусочно-линейными активационными функциями. Обучение МСП содержит два основных этапа: определение весов и определение порогов активационной функции. Обучение МСП для НСДГ подразумевает использование набора обучающих примеров, представляющих собой пары «матрица изображения - матрица контуров». Рассмотрим процесс формирования исходной системы линейных уравнений, используемой для определения вектора весов  $W$ . С помощью матрицы изображений формируется промежуточная матрица, количество строк которой равно количеству элементов матрицы описания исходного изображения (или его фрагмента), а количество столбцов зависит от размерности маски фильтра  $N$  и учитывает наличие постоянной составляющей. Порядок заполнения столбцов матрицы зависит от выбранного способа упорядочения компонент матрицы фильтра. С помощью матрицы контуров формируется вектор представления краев изображения. Таким образом, система уравнений, используемая для определения вектора весов  $W$  МСП, имеет вид:

$$S*W=F. \quad (7)$$

Размерность  $N*N+1$  вектора  $W$  равна количеству столбцов матрицы  $S$ . В прямоугольной матрице  $S$  количество строк превышает количество столбцов, следовательно, система (7) переопределена. Соответственно, задача определения вектора матрицы  $S$  формулируется в виде задачи минимизации евклидовой нормы невязки  $\min_w \|S*W - F\|$ .

Решение такой задачи требует использования регуляризации системы уравнений:

$$\min_w (\|S*W - F\| + \mu \|W\|). \quad (8)$$

Регуляризация позволяет найти решение с минимальной нормой вектора  $W$  за счет поиска согласованного минимума  $W$  нормы невязки для системы уравнений (8) и минимума нормы вектора, который входит в результирующий функционал с регуляризирующим коэффициентом  $\mu$ . Такая формулировка задачи обучения означает возможность использования одинаковых фрагментов изображений без риска получить вырожденную систему уравнений. С применением скалярного произведения задача (8) может быть представлена в виде:

$$\min_w ((S*W - F, S*W - F) + \mu(W, W)). \quad (9)$$

Решение задачи (9), позволяющее получить коэффициенты МСП, эквивалентно решению системы уравнений с квадратной матрицей размерности  $N*N+1$ :

$$W = (S' * S + \mu E)^{-1} S' * F, \quad (10)$$

где  $S'$  – транспонированная матрица  $S$ ;  $E$  – единичная матрица размерности  $N * N + 1$ .

При построении матрицы  $S$  и вектора  $F$  возможно как разбиение исходного изображения на фрагменты, так и использование нескольких пар обучающих примеров. Рассмотрим вариант обучения, связанный с последовательным использованием пар обучающих примеров. Пусть ИНС обучена с применением пары  $(S_1, F_1)$ . В этом случае значение вектора весов будет равно:

$$W1 = (S1' * S1 + \mu E)^{-1} S1' * F1. \quad (11)$$

В случае использования двух пар  $(S_1, F_1)$  и  $(S_2, F_2)$  вектор весов принимает вид:

$$W2 = (S1' * S1 + S2' * S2 + \mu E)^{-1} (S1' * F1 + S2' * F2). \quad (12)$$

Представление (12) может быть обобщено для  $k$  обучающих пар:

$$W(k) = (SE(k-1) + Sk' * Sk + \mu E)^{-1} (FE(k-1) + Sk' * Fk), \quad (13)$$

где  $SE(k) = \sum_1^k Sk' * Sk$ ,  $FE(k) = \sum_1^k Sk' * Fk$ .

#### 4. Программная реализация и оценка эффективности предложенного метода

Для тестирования разработанного метода использовались пространственные данные из базы данных ГИС экологического контроля, представленные растровой моделью [8].

В процессе тестирования осуществлялось имитационное моделирование процедуры 1 (предварительная нейросетевая обработка зашумленных изображений) и процедуры 2 (нейросетевое выделение контуров на изображении, предварительно улучшенном с помощью процедуры 1). Среднее время выбора структуры и обучения ИНС, осуществляющей предварительную обработку, составило около 85 секунд (на процессоре с частотой 4 ГГц). Среднее время обработки цветного изображения размером 512x512 пикселей составило около 1,2 секунды. Параметры алгоритма ГАИНС, используемого для выбора структуры ИНС: размер начальной популяции – 50; максимальное количество поколений (критерий останова) – 35.

При тестировании процедуры 2 для формирования обучающих примеров  $S_1$  и  $S_2$  были сгенерированы зашумленные изображения размерностью 100\*100 наложением шума заданной интенсивности на исходные изображения. Выделение контуров осуществлялось как с помощью синтезированного НСДГ, так и с помощью фильтров Собеля и Канни. Пример обработки реального зашумленного изображения с помощью процедуры 1 представлен на рис. 2 и 3.



Рис. 2. Пример улучшения качества изображений с использованием процедуры 1: исходное зашумленное изображение

Рис. 3. Пример улучшения качества изображений с использованием процедуры 1: улучшенное изображение

Пример выделения контуров зашумленных изображений, предварительно обработанных по процедуре 1, с использованием процедуры 2 представлен на рис. 4.

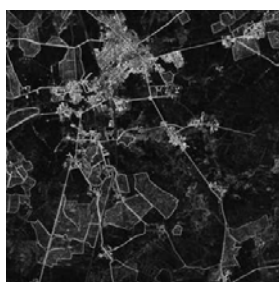
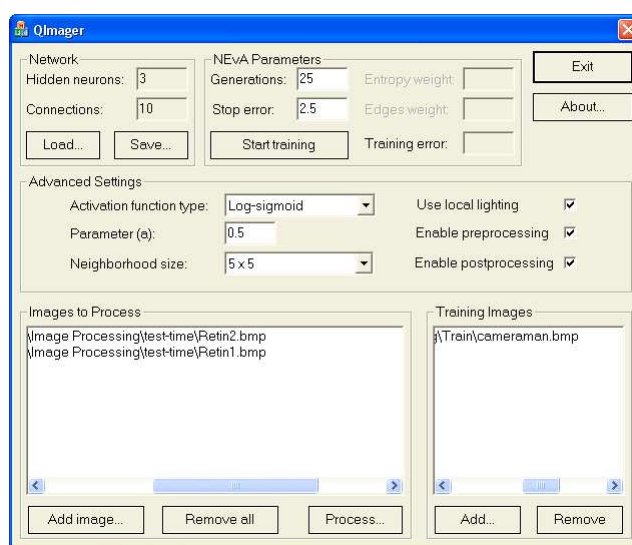
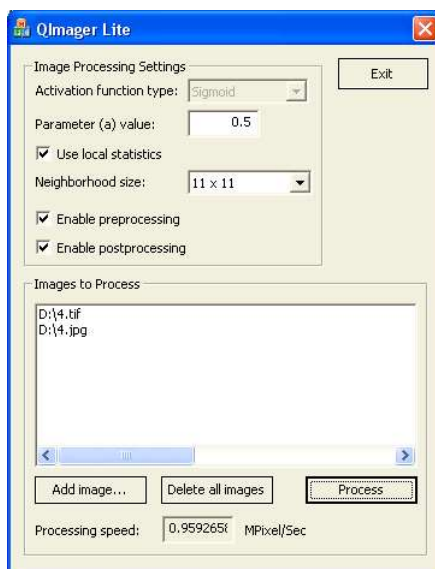


Рис. 4. Пример выделения контуров изображений с использованием процедуры 2

Было модифицировано программное обеспечение для обучения и тестирования ИНС в целях обработки изображений по процедурам 1 и 2. Элементы интерфейса программ для обучения ИНС и обработки изображений с использованием обученных ИНС представлены на рис. 5.



а



б

Рис. 5. Элементы интерфейса программ для обучения ИНС (а) и обработки изображений (б)

Результаты тестирования подтверждают перспективность предложенного метода для нейросетевой обработки фотореалистичных статических изображений в ГИС-приложениях.

### **5. Выводы и перспективы дальнейших исследований**

Предложенный метод позволяет осуществлять реализацию на параллельных вычислительных структурах процедур предварительной нейросетевой обработки зашумленных реалистических изображений (процедура 1) и последующего выделения границ (процедура 2). Тестирование метода подтвердило его работоспособность.

Выполненные исследования свидетельствуют об эффективности предложенного подхода и подтверждают целесообразность работ по созданию подсистемы нейросетевой обработки пространственных данных в составе ГИС. Для дальнейшего повышения эффективности метода целесообразно провести: исследование эффективности применения МСП-детекторов, обучаемых по различным алгоритмам фильтрации изображений; выбор зашумленных тестовых изображений, сгенерированных с использованием существующих баз пространственных данных; совершенствование процедур определения структур ИНС, применяемых в процедурах 1 и 2. Перспективным развитием этого подхода является рассмотрение возможности его практического применения в ГИС экологического мониторинга для сжатия и архивации пространственно-распределенных данных.

**Список литературы:** 1. Кобзаренко Д. Н. Особенности применения некоторых алгоритмов вычислительной геометрии в геоинформационных технологиях / Д. Н. Кобзаренко // Информационные технологии. 2011. № 8. С. 37-39. 2. Беленский Й. И. Метод выделения контура на слабоконтрастных размытых изображениях / Й. И. Беленский, И. В. Микулка // Вестник Винницкого политехнического института. 2012. № 3. С. 6-7. 3. Алиев М. В. Выделение контуров на мало контрастных и размытых изображениях с помощью фрактальной фильтрации / М. В. Алиев, А. Х. Панеш, М. С. Каспарьян // Вестник Адыгейского государственного университета. Серия 4: Естественно-математические и технические науки. 2011. № 3. С. 101-107. 4. Мишин А. Б. Фильтрация изображений с использованием нейросети [Текст] / А. Б. Мишин // «Телекоммуникации». 2012. № 13. С. 10-14. 5. Калинкина Д. Проблема подавления шума на изображениях и видео и различные подходы к ее решению / Д. Калинкина, Д. Ватолин. Москва: Техносфера, 2007. С. 118-128. 6. Gonzalez R. C. Digital Image Processing / R. C. Gonzalez, R. E. Wood // Prentice Hall. 2008. 954 p. 7. Шамраев А. А. Метод компьютерной обработки цифровых аэрофотоснимков / А. А. Шамраев, Е. О. Шамраева, О. Б. Дудинова // Системи обробки інформації. 2014. Вип. 7(123). С. 168-171. 8. Удовенко С. Г. Методы обработки картографических изображений / С. Г. Удовенко, А. А. Шамраев, Е. О. Шамраева // 23-я Международная конференция «СВЧ-техника и телекоммуникационные технологии» (КрыМиКо'2013). Севастополь: Вебер, 2013. С. 394-395.

*Поступила в редколлегию 12.12.2015*

**Удовенко Сергей Григорьевич**, д-р техн. наук, профессор кафедры электронных вычислительных машин ХНУРЭ. Адрес: Украина, 61166, Харьков, пр. Науки, 14, тел. 057-7021453, e-mail: serhii.udovenko@nure.ua.

**Дудинова Ольга Богдановна**, аспирантка кафедры электронных вычислительных машин ХНУРЭ. Адрес: Украина, 61166, Харьков, пр. Науки, 14.

## **ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ СИНТЕЗА СТРУКТУРЫ ПРОБЛЕМНО-ОРИЕНТИРОВАННЫХ ПРОГРАММНЫХ КОМПЛЕКСОВ**

---

Предлагается информационная технология синтеза проблемно-ориентированных программных комплексов, которая применяется для эволюционных моделей жизненного цикла программного обеспечения и обеспечивает конфигурирование и комплексирование программных средств с возможностью управления реализацией и восстановлением вычислительных процессов в параллельных и распределенных структурах вычислительных средств. Технология применяется в рамках процессов анализа требований к программным средствам, проектирования архитектуры, конструирования и комплексирования программных средств.

### **1. Введение**

В последнее время широко распространены такие технологии проектирования программного обеспечения (ПО) как SSADM, Meris, TDD, Gherkin [1-3]. Они позволили создать компонентно-ориентированные, сервисно-ориентированные методы, методы генерации, что обуславливается их эффективностью, малыми трудовыми и временными затратами для получения качественного программного продукта.

Однако существующие методы построения ПО имеют недостатки, связанные с отсутствием либо гибкой архитектуры программных средств (ПС), либо возможности динамического конфигурирования функциональности рабочих мест пользователей в условиях изменяющихся требований при использовании эволюционных жизненных циклов (ЖЦ) ПО [1,3]. При этом также не учитывается применение методов упрощения структуры ПК на основе анализа графовой модели его структуры в целях снижения временных затрат на проектирование и конфигурацию.

Одним из новых направлений в проектировании и разработке являются методы автоматизированного синтеза ПО с учетом меняющихся требований заказчика, главное преимущество которых – возможность предварительной оценки архитектурных и функциональных особенностей системы [4,5]. В этой связи является *актуальным* использование методов формализации гибкой архитектуры ПО с возможностью ее динамического изменения на основе графовой модели структуры ПС для решения задач в условиях изменяемых требований. Такие методы позволяют сформировать предварительное представление о системе до ее генерации, обеспечить динамическое конфигурирование и управление реализацией и откатом вычислительных процессов (ВП) для задач пользователей системы. Известные методы синтеза ПС используются, в основном, для формирования системы с компонентно-ориентированной или сервис-ориентированной архитектурой. Однако принципы и модели, которые лежат в основе метода формализации архитектуры ПО, можно адаптировать для решения задач, возникающих в процессе автоматизированного синтеза проблемно-ориентированных ПК. Поэтому *целью исследования* является создание информационной технологии (ИТ) синтеза структуры проблемно-ориентированных ПК с использованием графовых моделей структуры таких систем. *Задачами исследования* являются разработка методов формирования рациональной структуры проблемно-ориентированных комплексов и применение этих методов в предложенной ИТ синтеза структуры ПК.

### **2. Графовая модель структуры проблемно-ориентированного ПК в ярусно-параллельной форме**

Графовая модель представляет собой совокупность вершин, которым сопоставляются элементарные программные модули, выполняемые последовательно или независимо друг от друга. Ориентированные дуги между ними определяют тип связей по данным и описываются парой множеств характеристик программных модулей. Графовая модель формирует-

ся с избыточной для заданной предметной области (ПрО) функциональностью и обеспечивает гибкую конфигурацию системы в условиях эволюционно меняющихся требований.

Для построения модели требуются данные спецификаций требований к конечному программному продукту, информация, получаемая на этапах предпроектных исследований, существующие архитектурные решения, паттерны. Конечный вид графовой модели определяется выражением:

$$G = F(G_{исх}), \quad (1)$$

где  $F(G_{исх})$  определяет множество операций над исходной графовой моделью, которые позволяют привести ее к ярусно-параллельной форме с супервершинами, а  $G_{исх}$  является исходной графовой моделью вида:

$$G_{исх} = (V_{исх}, X_{исх}), \quad (2)$$

здесь  $V_{исх}$  – множество вершин  $v$ , которым сопоставляются программные модули (или же, в общем понимании, ВП), а  $X_{исх}$  – множество ориентированных дуг  $x_{ij} = (v_i, v_j)$  графа  $G_{исх}$ , определяющих зависимость по данным между программными модулями. Формирование множества вершин и дуг графа (2) происходит на основе выделения в ПрО соответствующих функциональных подсистем объекта исследования и установления потоков данных между ними. Для каждой  $i$ -й вершины модели (1) выделены два вектора данных:

$$D_i^{in} = \{d_1^{in}, d_2^{in}, \dots, d_k^{in}\} \quad (3)$$

– вектор входных данных, неизменяемых в процессе работы, и

$$D_i^{out} = \{d_1^{out}, d_2^{out}, \dots, d_k^{out}\}, \quad (4)$$

– вектор выходных данных, которые были модифицированы в процессе работы программного модуля. В случае изменяемых требований, определяющих модификацию архитектуры и расширение функциональности, графовая модель (1) дополняется подмножеством новых программных модулей  $V^H = \{v_i^H\}$  и подмножеством модифицируемых модулей  $V^M = \{v_i^M\}$ , позволяющих сконфигурировать структуру ПК для новой версии требований. Для получения новой версии структуры программы из графа (1) исключается подмножество вершин  $V^3 \subseteq V$ , которые заменяются подмножеством  $V^M$ , т.е. убираются программные модули устаревшей версии. Такая же операция продлевается с дугами  $X^3 \subseteq X$ . Это позволяет определить постоянную часть графа  $G: G^H = (V^H, X^H), G^H \subseteq G$ , где  $V^H = V \setminus V^3$ , а  $X^H = X \setminus X^3$ . Полученная графовая модель обладает следующими основными свойствами. На ней отсутствуют висячие вершины, для которых выполняется условие:

$$\forall v \in V, \{V' \subseteq V : |V'| > 1 : \exists v_i \in V' : \deg^-(v_i) = 0\}. \quad (5)$$

Для модели отсутствуют изолированные вершины, для которых выполняется условие:

$$\forall v \in V, \{\exists v_i \in V | \deg^+(v_i) = \deg^-(v_i) = 0\}. \quad (6)$$

Для нее отсутствуют парные дуги:

$$\forall i, j \in N, \{v_i, v_j \in V | \exists! x_{ij} \in X\} \quad (7)$$

и вершины с петлями:

$$\forall i \in N, \{v_i \in V | x_{ii} \notin X\} \quad (8)$$

Кроме этого, для ациклической графовой модели все вершины, для которых задан номер с учетом топологической сортировки и выполняется условие:

$$\begin{aligned} \forall i, j \in N, i < j, \{\exists v_i, v_j \in V : \mu[v_i, v_j] | \deg^-(v_i) = \deg^+(v_i) = \\ = \deg^-(v_{i+1}) = \deg^+(v_{i+1}) = \dots = \deg^-(v_j) = \deg^+(v_j) = 1\}, \end{aligned} \quad (9)$$

где  $\mu[v_i, v_j]$  – маршрут от вершины  $v_i$  к вершине  $v_j$ , объединяются в одну супервершину. Назначением предложенной графовой модели является описание структуры формируемого ПК, которая позволяет оценить архитектуру ПС до непосредственного комплексирования и

обеспечить динамическое конфигурирование рабочих мест пользователей, что является преимуществом предложенной модели. Приведение графовой модели к виду (2) для обеспечения выполнимости свойств (5)-(9) опирается на предложенный метод объединения вершин графовой модели структуры программного обеспечения информационной системы на основе оценки сложности и связности программных модулей [6].

### 3. Метод объединения вершин графовой модели структуры проблемно-ориентированного ПК

Данный метод предназначен для сокращения временных затрат на разработку системы путем объединения вершин в супервершины на базе модифицированного алгоритма Косарайо, дополнительно обрабатывающего графовую модель на основе комплексного критерия оценки эффективности структуры ПК [6]:

$$K_c = \frac{K_{\text{общ}}}{K'_{\text{общ}}} \cdot \frac{K'_{\text{слож}}}{K_{\text{слож}}}, \quad (10)$$

где  $K'_{\text{слож}}$  и  $K'_{\text{общ}}$  – значения среднего коэффициента сложности и обобщенного критерия, полученные после оптимизации графовой структуры, а  $K_{\text{слож}}$  и  $K_{\text{общ}}$  – значения критериев до оптимизации.

Метод объединения вершин графовой модели структуры проблемно-ориентированных ПК включает следующие этапы:

Этап 1. Устранение топологических некорректностей графовой модели в соответствии с формулами (5)-(6).

Этап 2. Получение для каждого  $i$ -го программного модуля количества вызываемых модулей, количества элементов и структур данных, обновляемых  $i$ -м модулем, и значения коэффициента системной сложности.

Этап 3. Расчет среднего коэффициента сложности  $K_{\text{сложн}}$  (10).

Этап 4. Расчет значения критериев  $K_{\text{общ}}$  и  $K_{\text{сложн}}$  и получение значения комплексного критерия оценки эффективности структуры программного средства  $K_c$  (10) для заданной версии требований к конфигурации системы.

Этап 5. Поиск компоненты сильной связности для заданного графа и конденсация найденной компоненты.

Этап 6. Поиск пары вершин по условию (7) после того, как на графе не осталось ни одной компоненты сильной связности (для любых двух вершин  $v_i$  и  $v_j$  невозможно найти двух одновременно существующих ориентированных путей  $\mu[v_i, v_j]$  и  $\mu[v_j, v_i]$ ).

Этап 7. Вычисление значения комплексного критерия оценки эффективности структуры программного средства (10) для найденной пары вершин.

Этап 8. Объединение пары вершин в одну супервершину, если значение критерия строго меньше единицы, и повторение с шага 3 до тех пор, пока значение комплексного критерия (14) не будет удовлетворять конфигурационным требованиям.

Для предложенного метода проведена оценка сложности, которая совпадает с алгоритмом Косарайо и соответствует величине  $O(n)$ . Преимуществом предложенного метода является то, что он позволяет уменьшить временные затраты на конфигурирование путем упрощения структуры графовой модели, опираясь на ее топологические особенности и конфигурационные требования к системе.

Графовая модель структуры проблемно-ориентированного ПК, обработанная на основе метода объединения вершин, подзвояет описать статический аспект формируемой ПС. Для оценки поведенческих свойств системы проверки нефункциональных к ней требований необходимо построение автоматной модели комплекса на основе автоматного метода.

#### 4. Автоматный метод оценивания последовательности взаимодействия модулей проблемно-ориентированного ПК

Автоматный метод используется для проектирования исполнителя ВП, моделируя сценарии взаимодействия программных модулей и сравнивая результаты моделирования с нефункциональными требованиями к ПК. Он применяет графовую модель структуры проблемно-ориентированного ПК, для которой определены вершины, отвечающие за формирование пользовательского диалога и за формирование соответствующих им подграфов модулей, реализующих связанные задачи системы. Метод основан на модели конечного автомата (КА), которая задается стандартным набором элементов:

$$A = \{S, X, Y, s_0, \delta, \lambda\},$$

где  $S$  – конечное непустое множество состояний;  $X$  – конечное непустое множество входных сигналов (входной алфавит);  $Y$  – конечное непустое множество выходных сигналов (выходной алфавит);  $s_0 \in S$  – начальное состояние;  $\delta : S \times X \rightarrow S$  – функция переходов;  $\lambda : S \times X \rightarrow Y$  – функция выходов. При этом множества состояний, переходов, входных и выходных сигналов задаются графически с помощью диаграммы переходов, а структура конечных автоматов, взаимосвязи между вложенными автоматами – с помощью диаграммы классов. Для реализации задач прямого (автомат AForwardSM) и обратного (восстановление результатов – автомат AbackwardSM) управления ходом ВП предложено использовать основной управляющий КА, который обеспечивает взаимодействие двух вложенных КА. На основании определенной структуры сформулированы основные задачи, реализуемые предложенными КА. Для управляющего КА это:

- 1) формирование подграфа задачи;
- 2) проверка состояния задачи;
- 3) проверка результата на корректность;
- 4) изменение состояния задачи.

Для автомата AForwardSM это:

- 1) запуск ВП;
- 2) архивация данных ВП.

Для автомата AbackwardSM это восстановление архивных данных для каждой вершины соответствующих ярусов.

Предложенный метод включает в себя следующие этапы:

Этап 1. Выделение базовых состояний ПК и условий переходов на основе входных и выходных данных вершин графовой модели.

Этап 2. Формирование структуры управляющего КА и его вложенных подавтоматов, реализующих поведение системы, с учетом данных спецификаций вершин графовой модели.

Этап 3. Определение принципов взаимодействия КА с программными модулями на основе архитектурных шаблонов объектно-ориентированного программирования для организации распределенного или параллельного выполнения программы.

Этап 4. Формирование исходной информации, требуемой для организации работы КА, такой как матрицы смежности графовой модели структуры проблемно-ориентированного ПК и транспонированной матрицы смежности для управления прямым и обратным ходом ВП.

Этап 5. Разработка структуры будущего ПК с учетом взаимосвязи графовой модели и управляющих конечных автоматов.

Этап 6. Генерация ПК на основе полученной структуры.

Предложенные автоматные модели обладают следующими обязательными свойствами. Для автомата AForwardSM: 1) ВП  $P(v_i)$  для заданной  $i$ -й вершины будет запущен только по запросу пользователя; 2) гарантия того, что процесс  $P(v_i)$  обязательно когда-нибудь будет запущен, если не будет ошибок вычислений; 3) гарантия того, что процесс  $P(v_i)$  будет запущен только тогда, когда вершина  $v_i$  находится в состоянии «не запущен» или «выполнен»; 4) вершина  $v_i$  когда-нибудь обязательно будет находиться в состоянии «не запущен» или «выполнен»; 5) для каждой необходимой вершины графовой модели

будет выполнен процесс архивации данных; 6) для последовательности ВП на основе существующей ЯПФ графовой модели можно выделить такой ВП, который становится активным только тогда, если все предшествующие процессы завершат свою работу; 7) для всего множества ВП программной системы можно выделить такое подмножество ВП, которые размещены на одном ярусе ЯПФ графовой модели и не имеют информационных зависимостей друг от друга, что позволяет запускать их одновременно.

Для автомата ABackwardSM: 1) восстановление данных ВП (backtracking) будет выполнено только по запросу пользователя; 2) восстановление данных ВП будет выполнено только при наличии архива данных для каждой вершины подграфа ВП; 3) для каждой вершины выделенного подграфа произойдет восстановление данных только в состоянии, если другой пользователь не задействовал эту вершину для себя (статус «не запущен») и если восстановление данных не происходило вовсе (статус «не восстановлен»), или данные уже были восстановлены ранее (статус «восстановлен»).

На основе определенных и обозначенных состояний КА (табл. 1) и событий, по которым КА переходит в эти состояния (табл. 2), формализуются свойства модели на базе темпоральной логики LTL, которые отвечают требованиям к поведению ПС и дополняются новыми свойствами в случае появления очередных версий нефункциональных требований.

Таблица 1

Переходы конечного автомата

Обозначение	Наименование
e1	получено разрешение
e5	ошибка выполнения
e6	номер вершины меньше максимального для текущего яруса
e7	номер текущего яруса меньше количества ярусов
e16	заданы условия и параметры ВП
e43	данные считаны
e81/ e82	архив существует / отсутствует
e101/ e102	данные архива не восстанавливались / восстанавливаются
e103	данные архива восстановлены
e121/e122/e123	ВП не запущен / выполняется / выполнен

Таблица 2

Состояния конечного автомата

Обозначение	Наименование
o3.zf22	инициализация ВП для автомата AForwardSM
o3.zf24	переход к следующему ВП
o3.zf28/ o3.zf31	запуск ВП/ ожидание завершения ВП
o3.zf35/ o3.zf36	архивация данных/получение данных
o2.zf45	восстановление данных для автомата ABackwardSM

Для автомата AForwardSM это:

- 1)  $\neg(\neg e1 U o3.zf28)$  ;
- 2)  $\neg(e5 U o3.zf28)$  ;
- 3)  $\neg(\neg(e121 \vee e123) U o3.zf28) = \neg(\neg e121 \wedge \neg e123) U o3.zf28$  ;
- 4)  $F o3.zf31$  ;
- 5)  $e43 F o3.zf35$  .

Свойство 6) для автомата AForwardSM формализовано следующим образом:

$$F_{\text{общ}}^{\text{AV}} = \neg(F_1^{\text{AV}} \wedge F_2^{\text{AV}} \wedge \dots \wedge F_n^{\text{AV}}) = \neg F_1^{\text{AV}} \vee \neg F_2^{\text{AV}} \vee \dots \vee \neg F_n^{\text{AV}} ,$$

где  $F^{AV} = \neg(\neg(e122Uo3.zf31) \wedge \neg(e121Uo3.zf22)U(\neg(\neg e16Uo3.zf28)))$ , поскольку в том случае, когда существует  $n$  вершин, таких что  $\forall k = \overline{m, n}; \exists x_{ki} = (v_k, v_i)$ , где  $m$  и  $n$  целые числа и полустепень захода  $\deg^+(v_i) = n$ , общее логическое высказывание заключается в одновременном выполнении этого свойства для всех  $n$  дуг.

Свойство 7) для автомата AForwardSM формализовано таким образом:

$$F_{\text{общ}}^{PV} = \neg(F_1^{PV} \wedge F_2^{PV} \wedge \dots \wedge F_n^{PV}) = \neg F_1^{PV} \vee \neg F_2^{PV} \vee \dots \vee \neg F_n^{PV},$$

где

$$F^{PV} = \neg(\neg(\neg(\neg(e6 \wedge e7)Uo3.zf28))U(\neg(\neg e122U(o3.zf24 \wedge o3.zf36)))) = \neg((\neg e6 \vee \neg e7)Uo3.zf28U(\neg(\neg e122U(o3.zf24 \wedge o3.zf36))))),$$

так как, если задать подмножество вершин  $v_i \in V_r$ , где  $i = \overline{m, n}$ , принадлежащих одному ярусу и между которыми не существует дуги, то в общем случае логическая формула также будет выглядеть, как конъюнкция формул, соотнесенных с заданными вершинами.

Для автомата ABackwardSM свойства формализованы таким образом:

- 1)  $\neg(\neg e1Uo2.zf45)$ ;
- 2)  $\neg(\neg e82Uo2.zf45) \vee \neg(\neg e81Uo2.f45)$ ;
- 3)  $\neg(\neg(e121 \wedge e101 \vee e103)Uo2.zf45) \vee \neg((e122 \wedge e102)Uo2.f45) = \neg((\neg e121 \vee \neg e101 \wedge \neg e103)Uo2.zf45) \vee \neg((e122 \wedge e102)Uo2.f45)$ .

Сформулированные свойства являются основой для определения правил исполнения ВП, поэтому автоматная модель, разработанная в соответствии с предложенным методом, позволяет сформировать ограничения к структуре ПК с учетом меняющихся требований и повысить надежность его функционирования, опираясь на данные графовой модели структуры проблемно-ориентированного ПК, что отличает метод от технологии BDD.

### 5. ИТ конфигурирования проблемно-ориентированного ПК в условиях меняющихся требований

ИТ конфигурирования проблемно-ориентированных ПК в условиях меняющихся требований позволяет отразить требования текущей версии на графовой модели и обеспечить переконфигурацию модулей проблемно-ориентированных ПК. Технология включает в себя следующие этапы (рис.1).

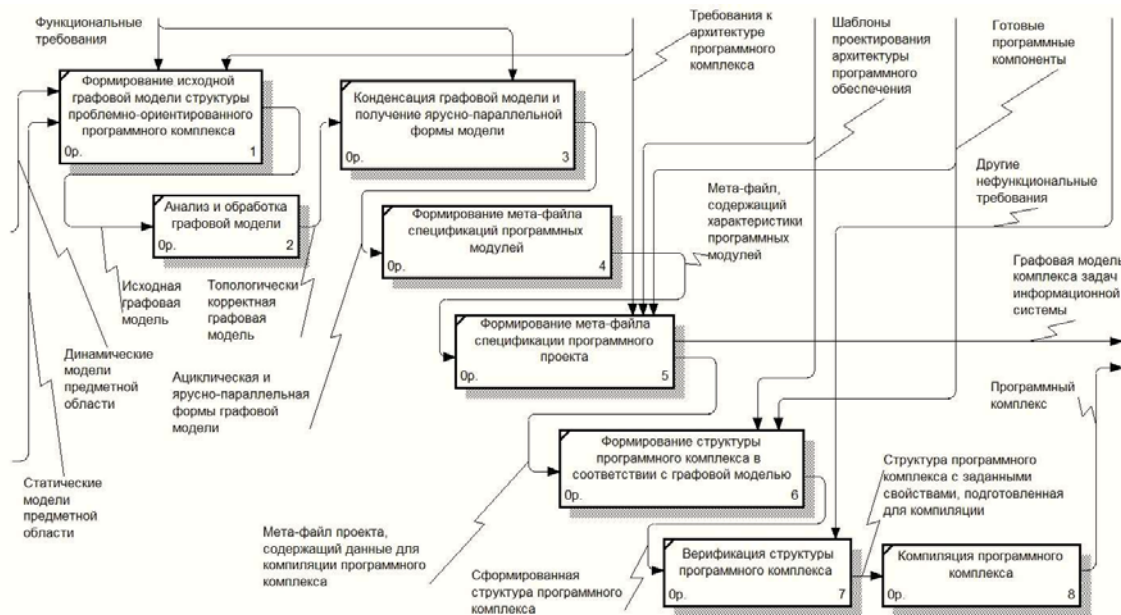


Рис. 1. Этапы информационной технологии синтеза проблемно-ориентированных ПК

Этап 1. Формирование исходной графовой модели структуры проблемно-ориентированного ПК.

Этап 2. Анализ и обработка графовой модели.

Этап 3. Конденсация и получение ЯПФ модели.

Этап 4. Формирование мета-файла спецификаций модулей.

Этап 5. Формирование мета-файла спецификации проекта.

Этап 6. Формирование структуры ПК в соответствии с графовой моделью.

Этап 7. Верификация структуры ПК.

Этап 8. Компиляция ПК.

Первый этап информационной технологии обеспечивается информацией из доступных документов, описывающих требования заказчика: первичные спецификации, диаграммы статического и динамического моделирования ПрО. На основе функциональных и архитектурных требований к ПК для сущностей модели ПрО определяются функции объектов, формирующих графовую модель, которая на втором этапе проверяется на топологическую корректность. На третьем этапе на основе требований заказчика выполняется метод объединения вершин графовой модели и формируется рациональная структура графа. На четвертом этапе на базе полученной структуры происходит описание программных модулей при помощи мета-файлов спецификаций. На пятом этапе с использованием архитектурных шаблонов формируется необходимый для генерации ПК и динамической конфигурации мета-файл программного проекта, который описывает принципы взаимодействия модулей. Данный файл является входным для процесса проверки выполнения нефункциональных требований на основе оценки верификации последовательности взаимодействия модулей на седьмом этапе, после успешного окончания которого выполняется генерация комплекса и отчуждение графовой модели для конечного пользователя сформированной системы. В случае новой версии требований процесс повторяется с третьего этапа.

Информационная технология позволяет осуществить синтез графовой модели структуры проблемно-ориентированного ПК и динамически выполнить конфигурацию его архитектуры и функциональности рабочих мест в условиях меняющихся требований заказчика. При проверке применимости предложенной ИТ для синтеза структуры ПК «Автоматизированная система подготовки производства сложных электронных устройств» по предоставленной информации о ПрО был рассчитан процент ручной настройки системы, необходимой для динамического конфигурирования. Расчет выполнен на основе ЛОС-оценок программного кода и сравнивался со стандартным методом конфигурирования системы на основе данных отчета по внедрению ERP-систем в производство за 2015 год [7], согласно которому большинство предприятий (в среднем 44% предприятий всех отраслей промышленности и 38% предприятий сферы машиностроения) требуют от 26 до 50% изменения исходного кода в соответствии с настройками и требованиями заказчиков системы (рис. 2). Предложенная технология позволила сократить процент ручной работы персонала до 10% для модификации исходного кода традиционным способом в размере 26 и до 18% для соответствующей границы в 50%, т.е. предложенная информационная технология значительно снижает временные затраты на переконфигурацию ПК – на 36 %.

## **6. Выводы**

ИТ конфигурирования проблемно-ориентированных ПК в условиях изменяющихся требований в отличие от существующих технологий дополнена этапами анализа и обработки модели ПрО в целях синтеза графовой модели проблемно-ориентированного ПК, конфигурирования этой структуры с учетом текущих функциональных требований и проверки выполнения нефункциональных требований на основе оценивания последовательности взаимодействия модулей в рамках процессов анализа требований, проектирования архитектуры, конструирования и комплексирования ПК, что обеспечивает значительное снижение стоимости изменений программного обеспечения. Предложенная технология позволяет снизить временные затраты на конфигурирование программного обеспечения на 36 % и применяется как дополнение к существующим технологиям проектирования и разработки программного обеспечения, подразумевая ее использование для эволюционных ЖЦ таких технологий как BDD, TDD и других в рамках процессов анализа требований к программ-

ным средствам, проектирования архитектуры, конструирования и комплексирования программных средств.

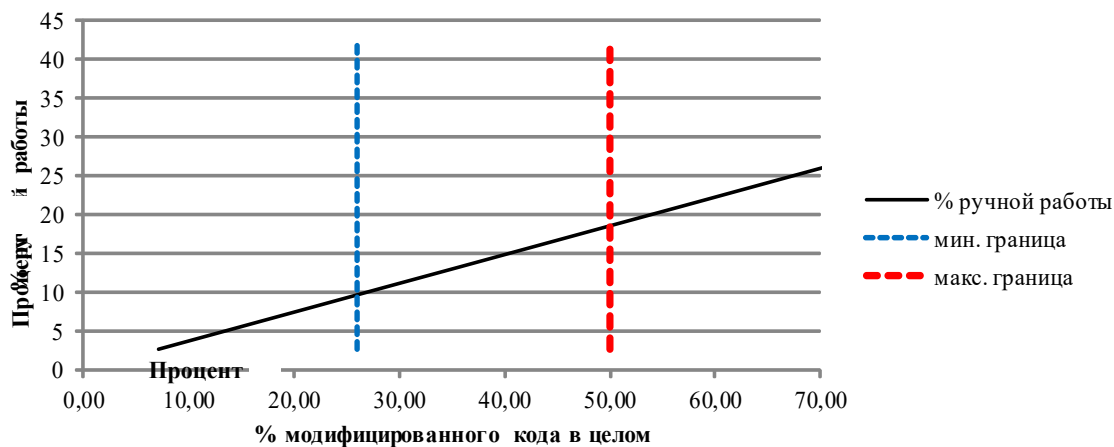


Рис. 2. Доля ручной настройки системы относительно процента изменения исходного программного кода

Определение принципов управления, отката и синхронизации ВП в параллельных вычислительных средах, а также методов формирования гибкой архитектуры программных комплексов с ее адаптацией для сервис-ориентированной и облачной архитектур является предметом дальнейших исследований.

**Список литературы:** 1. *Amodeo E.* Learning Behavior-driven Development with javascript + Code / Enrique Amodeo. Birmingham:Packt Publishing. 2015. 392 p. 2. *Бек К.* Экстремальное программирование: разработка через тестирование. Библиотека программиста. СПб.: Питер, 2003. 224 с. 3. *Smart J. F.* BDD in Action: Behavior-driven development for the whole software lifecycle / JOHN FERGUSON SMART / Publisher: Manning Publications, Shelter Island, NY. 2015. 384 p. 4. *Лаврищева Е.М.* Сборочное программирование. Основы индустрии программных продуктов/ Е.М. Лаврищева, В.Н. Грищенко. Киев: Наук. думка. 2009. 372 с. 5. *Лаврищева Е.М.* Software Engineering компьютерных систем. Парадигмы, технологии и CASE-средства программирования / Е. М. Лаврищева. К.: Наук. думка. 2013. 283 с. 6. *Солодовников А.С.* К вопросу оценивания эффективности и сложности структуры программного средства / А.С. Солодовников // Проблеми інформаційних технологій. 2014. №2 (16). С.125-129. 7. The 2015 Manufacturing ERP Report / Panorama Consulting Solutions. Denver, Colorado. 2015. 13 p. [https://cdn2.hubspot.net/hubfs/240423/Whitepapers/2015\\_Pan\\_Man\\_ERP\\_Report.pdf](https://cdn2.hubspot.net/hubfs/240423/Whitepapers/2015_Pan_Man_ERP_Report.pdf)

Поступила в редколлегию 07.12.2015

**Чайников Сергей Иванович**, канд. техн. наук, профессор кафедры системотехники ХНУРЭ. Адрес: Украина, 61166, Харьков, пр. Науки, 14.

**Солодовников Андрей Сергеевич**, ассистент кафедры биологической и медицинской физики и медицинской информатики, Харьковский национальный медицинский университет, тел. 066-774-773-1, e-mail: andrey.sldv@rambler.ru

## МЕТОД АВТОМАТИЧЕСКОГО ПОСТРОЕНИЯ ОНТОЛОГИЧЕСКИХ МОДЕЛЕЙ С ДРЕВОВИДНОЙ СТРУКТУРОЙ КОНЦЕПТОВ

Предлагается метод автоматического построения онтологической модели данных на основе анализа корпуса научных текстов для определенной предметной области, что позволяет формировать онтологию с древовидной структурой концептов с учетом семантических связей между ними. Осуществляется программная реализация метода, предусматривающая возможность последовательного построения древовидной онтологической модели по исходному корпусу монотематических текстов.

### 1. Постановка задачи

Онтологии предметной области в настоящее время широко применяются в области построения поисковых систем, систем представления знаний, инженерии знаний и при решении задач семантической интеграции информационных ресурсов. Под онтологией понимается «формальная спецификация концептуализации, которая имеет место в некотором контексте предметной области» [1]. В свою очередь, концептуализация представляет собой представление предметной области через описание множества понятий (концептов) предметной области и связей (отношений) между ними. В последние годы сформировалась парадигма компьютерных онтологий, основными признаками которых являются [2]:

- иерархическая структура конечного множества понятий, описывающих заданную предметную область;
- структура онтологии может быть представлена онтографом, вершинами которого являются понятия (концепты), а дугами – семантические отношения (связи) между ними;
- концепты и связи между ними интерпретируются в соответствии с результатами анализа электронных источников знаний (например, корпуса текстов) заданной предметной области;
- онтограф должен быть представлен формально на одном из языков описания онтологий.

В общем случае компьютерная онтология некоторой предметной области формально представляется упорядоченной тройкой [3]:

$$O = \langle X, R, F \rangle, \quad (1)$$

где  $X, R, F$  – конечные множества соответственно:  $X$  – концепты (понятия, термины) предметной области;  $R$  – отношения между ними;  $F$  – функции интерпретации  $X$  и/или  $R$ .

Рассмотрение граничных случаев множеств в (1):  $R = \emptyset$ ;  $R \neq \emptyset$ ;  $F = \emptyset$ ;  $F \neq \emptyset$  во всех четырех комбинациях значений  $R$  и  $F$  дает различные варианты онтологических конструкций, начиная от простого словаря и таксономии до формальной структуры концептуальной базы знаний для высокоинтеллектуальных знание-ориентированных систем [4].

По своей функциональной полноте и степени формальности различают три вида онтологий: простая, полная (или строгая) и множество промежуточных или неполных онтологий.

Простая – это такая онтология, в которой  $R = \emptyset$ ;  $F = \emptyset$ . Она служит (в основном) для однозначного восприятия научным сообществом понятий в соответствующей прикладной области.

Строгая или полная ( $R \neq \emptyset$ ,  $F \neq \emptyset$ ) – это такая онтология, в которой множества концептов и концептуальных отношений являются максимально полными, а к функциям интерпретации добавляются аксиомы, определения и ограничения. При этом описания всех компонент представлены на некотором формальном языке, доступном для их интерпретации компьютером. Схема формальной модели полной онтологии описывается четверкой:

$$O = \langle X, R, F, A(D, R_s) \rangle, \quad (2)$$

где  $X$  – множество концептов;  $R$  – множество концептуальных отношений между ними;  $F: X \times R$  – конечное множество функций интерпретации, заданных на концептах и/или отношениях;  $A$  – конечное множество аксиом, которые используются для записи всегда истинных высказываний (определений и ограничений);  $D$  – множество дополнительных определений понятий;  $R_s$  – множество ограничений, определяющих область действия понятийных структур.

В промежуточных или неполных онтологиях ( $R = \emptyset, F \neq \emptyset; R \neq \emptyset, F = \emptyset$ ) для каждого концепта (или их большей части) добавлены аксиомы и определения, представленные на естественном языке (ЕЯ). Одним из распространенных вариантов неполной онтологии является структура  $O = \langle X, R \rangle$ , где множество  $F$  в явном виде отсутствует ( $F = \emptyset$ ), в предположении, что концепты общеизвестны (определены по умолчанию) либо (и) достаточно полно интерпретированы отношениями  $R$ .

Применение онтологического подхода для автоматической обработки текстов на естественном языке предполагает сопоставление понятиям онтологии предметной области (к которой принадлежит множество текстов) языковых выражений (слов и словосочетаний), которыми понятия могут быть выражены в текстах. При этом структура концептов такой онтологии, представленных словосочетаниями из корпуса текстов, является древовидной, причем конкретный вид дерева определяется сложностью словосочетания.

Качество формируемых онтологий, используемых для создания поисковых систем, во многом определяется полнотой учета в онтологической модели наиболее значимых концептов для корпуса анализируемых текстов с учетом их тематической специфики (под концептами будем в дальнейшем понимать наиболее значимые слова и словосочетания в анализируемом тексте, которые могут быть учтены в онтологической модели). В связи с этим целесообразно решить задачу формирования множества концептов будущей онтологии типа  $O = \langle X, R \rangle$  с учетом связей между ними (при возможности оценивания концептов по двум атрибутивным показателям, связанным с анализом исходного корпуса текстов предметной области: TF и TF/IDF). Процедура определения шаблонных связей типа «часть-целое» была рассмотрена в работе [5]. Более сложной является задача нахождения связей типа «отношение» между выбранными концептами. В значительной степени именно задание таких отношений наделяет онтологию интеллектуальным смыслом, что позволяет компьютеру, работающему с такой онтологией, максимально эффективно использовать заложенные в нее знания. Следует отметить, что при поиске слов и словосочетаний, которые могут применяться в качестве концептов, сформированное множество концептов-претендентов не всегда соответствует такому же множеству, составленному экспертом предметной области. Это приводит к тому, что некоторые важные понятия предметной области могут не попасть в автоматически создаваемую онтологию. Кроме того, в этих методах отсутствует процедура общего ранжирования по значимости списка всех концептов-претендентов, а осуществляется лишь раздельное ранжирование слов и словосочетаний, входящих в этот список.

В частности, возникают следующие проблемы: не всегда удастся правильно найти связи между концептами; не всегда удастся выделить концепты, имеющие связь с наибольшим количеством других концептов; найденные связи между концептами будущей онтологии не всегда актуальны для конкретной предметной области. При этом не только повышается используемый объем памяти и увеличивается время на создание онтологии и обработку запросов к ней, но и избыточным становится объем онтологии, что снижает оперативность дальнейшего ее применения. Целесообразно рассмотреть возможность устранения перечисленных трудностей на основе комбинированного применения и модификации существующих методов определения релевантных связей между концептами формируемых онтологических моделей.

*Целью* данного исследования является разработка и программная реализация метода автоматического построения онтологий с древовидной структурой концептов-словосочетаний для заданной предметной области по корпусу текстов.

## 2. Метод построения онтологии

Концепция предлагаемого метода автоматического поиска древовидной структуры концептов формируемой онтологии состоит в реализации следующих этапов:

Этап 1. Выделение концептов высшего уровня онтологического дерева.

Этап 2. Выделение ключевых слов в концептах высшего уровня.

Этап 3. Установление связей между концептами высшего уровня.

Этап 4. Определение дочерних концептов.

Этап 5. Установление связей между концептами онтологии.

Этап 6. Построение онтографа.

Очевидно, что количество дочерних концептов для каждого из концептов высшего уровня может быть различным и общая структура онтологического дерева будет зависеть от предметной области и репрезентативности исходного корпуса текстов.

Рассмотрим подробнее этапы предлагаемого метода.

**Выделение концептов высшего уровня онтологического дерева.** Выделение концептов высшего уровня (КВУ) для любой онтологии предметной области (ОПО) является важным этапом в общем алгоритме проектирования, так как древовидные ОПО строятся фрагментарно, начиная от исходной вершины, и именно качество задания КВУ определяет в дальнейшем эффективность объединения концептов и связей вершин в общую онтологию. Очевидно, что концепты низших (дочерних) уровней ОПО являются подклассами КВУ и поэтому наследуют признаки понятия-класса, если они связаны между собой отношениями частичного порядка.

Под концептами высшего уровня формируемой онтологии будем понимать наиболее важные (значимые) для рассматриваемой предметной области словосочетания  $W_i$ ,  $i = \overline{N}$  ( $N$  – число концептов высшего уровня) из исходного корпуса текстов. Эти словосочетания являются элементами концептуальной составляющей формируемой онтологии вида  $O = \langle X, R \rangle$ , где  $X = W$ .

В работе [6] был предложен метод поиска таких словосочетаний, основанный на количественной оценке важности элементов текста с использованием значений TF, TF/IDF и так называемых рангов слов. Исходной операцией здесь является предварительное упорядочение по убыванию важности и составление соответствующих списков важных слов для показателей TF и TF/IDF. Под частными рангами  $R_1$  и  $R_2$  некоторого слова понимаются значения величин, обратных номерам позиции этого слова в упорядоченных списках для TF и TF/IDF соответственно. Под общим рангом  $R$  некоторого слова будем понимать коэффициент, который соответствует наибольшему из значений частных рангов ( $R_1$  и  $R_2$ ) этого слова для анализируемого корпуса текстов.

Предлагается оценивать в анализируемых текстах значимость слов  $w_i$  в  $i$ -м тексте корпуса по значениям коэффициента  $K(w_i)$ , рассчитываемого по следующей зависимости:

$$K(w_i) = \frac{1}{|R_1 - R_2|} * \frac{Q(w_i)}{N_{\max} - N_{\min}}, \quad (3)$$

где  $Q(w_i)$  – количество текстов, в которых содержится слово  $w_i$ ;  $N_{\max}$ ,  $N_{\min}$  – наибольшее и наименьшее число вхождений слов  $w_i$  в корпус текстов соответственно.

Согласно (3) значимые термины, выделяемые для создания отношений между словами-концептами, должны встречаться в большинстве текстов корпуса и при этом быть максимально равномерно распределены в каждом тексте корпуса текстов.

После определения значений  $R_1$  и  $R_2$  производим модификацию исходных списков. При этом слово относим к первому модифицированному списку, если для него  $R_1 > R_2$ , и, соответственно, ко второму, если  $R_2 > R_1$ .

Очевидно, что важность словосочетаний для текущего текста зависит от наличия в них слов каждого из модифицированных списков. Например, словосочетание, полностью составленное из слов первого списка, недостаточно хорошо отображает семантику текста,

так как все слова в нем, скорее всего, окажутся слишком общими. Однако, если словосочетание составлено полностью из слов второго списка, в нем могут отсутствовать ключевые слова, имеющие максимальную частоту вхождения в документы анализируемого корпуса текстов.

Важно также правильно выбрать максимальную длину оцениваемых словосочетаний. Очевидно, что с ее увеличением уменьшается вероятность присутствия в тексте осмысленных словосочетаний заданной длины.

В связи с этим представляется целесообразным в общий критерий количественной оценки важности словосочетаний ввести нормированный коэффициент  $K(W_i)$ , который будет зависеть как от количества вхождений в словосочетание  $w_i$  слов из первого и второго модифицированных списков, так и от длины словосочетания.

Анализ списков словосочетаний (для представительного набора текстов по направлению «Компьютерные науки») позволил экспериментально оценить влияние длины словосочетаний, а также количества слов из первого и второго модифицированных списков на вероятность присутствия этих словосочетаний в документах анализируемого корпуса текстов. Очевидно, что такая вероятность может быть принята в качестве коэффициента  $K(W_i)$ , значения которого находятся в диапазоне  $[0; 1]$ .

В таблице представлены значения коэффициента  $K(W_i)$  для разных типов словосочетаний. Предлагается оценивать важность словосочетания, перемножая ранги отдельно взятых слов. При этом, поскольку ранги слова нормированы от нуля до единицы, то с увеличением длины словосочетания уменьшается результат такого произведения. В связи с этим его необходимо умножить на количество слов в рассматриваемом словосочетании.

Следует также учитывать, что слова, для которых  $R_1 > R_2$ , определяют специфику анализируемого текста, т.е. слова из первого списка с большой вероятностью попадут в большинство словосочетаний, характерных для этого текста. Следовательно, оценки важности таких словосочетаний будут близки. Наиболее же специфичными для конкретного текста являются слова второго списка, для которых  $R_2 > R_1$ , следовательно, именно за счет этого можно повысить релевантность критерия оценки важности словосочетания.

Значения коэффициента  $K(W_i)$  для разных типов словосочетаний

№	Количество слов в словосочетании по первому списку	Количество слов в словосочетании по второму списку	$K(W_i)$
1.	2	0	0,3
2.	1	1	1
3.	3	0	0,2
4.	2	1	0,6
5.	1	2	0,7
6.	3	1	0,5
7.	1	3	0,8

Кроме того, величина модуля разности рангов слов по разным спискам ( $|R_{1j} - R_{2j}|$ ) влияет на вероятность того, что слово, которому соответствует максимальный ранг, является важным для анализируемого корпуса текстов: чем больше минимальный модуль разности двух рангов одного и того же слова, тем выше находятся все слова из словосочетания в каком-либо из двух списков и тем важнее словосочетание в целом.

Для комплексной оценки важности словосочетания  $W_i$  в рассматриваемом тексте можно использовать следующий коэффициент  $M(W_i)$ :

$$M(W_i) = K(W_i) * \left( \prod_{i=1}^n R_i \right) * \min(|R_{1j} - R_{2j}|) * n * \max(TF / IDF_j); j = 1, \dots, n, R_{1j} \neq R_{2j}, \quad (4)$$

где  $n$  – количество слов в словосочетании  $W_i$  (не считая стоп-слов); TF/IDF – коэффициенты оценки важности слова;  $R_i$ ,  $R_{1j}$ ,  $R_{2j}$  – общий и частные ранги слова  $w_j$  из словосочетания  $W_i$  соответственно.

Важность словосочетаний  $M(W_i)$  предлагается оценивать по произведению рангов соответствующих слов. При этом, поскольку ранги слова нормированы от нуля до единицы, с увеличением длины словосочетания уменьшается результат такого произведения. В связи с этим его необходимо умножить на количество слов в рассматриваемом словосочетании. Следует также учитывать, что слова, для которых  $R_1 > R_2$ , определяют специфику анализируемого текста, т.е. слова из первого списка с большой вероятностью попадут в большинство словосочетаний, характерных для этого текста. Наиболее же специфичными для конкретного текста являются слова второго списка, для которых  $R_2 > R_1$ , следовательно, именно за счет этого можно повысить релевантность критерия оценки важности словосочетания.

Общий алгоритм поиска наиболее важных словосочетаний (для заданного корпуса текстов) состоит в следующем:

Шаг 1. Предварительно ранжируются по важности исходные списки слов по заданному корпусу электронных текстов (по  $R_1$  и  $R_2$ ) и определяются их общие ранги.

Шаг 2. Производится модификация ранжированных списков по приведенным выше правилам.

Шаг 3. Формируется исходный список словосочетаний из слов модифицированных списков.

Шаг 4. Определяется оценка важности словосочетаний  $M(W_i)$ .

Таким образом, в концептах высшего уровня могут быть использованы словосочетания с наибольшей оценкой  $M(W_i)$  либо отдельные слова из этих словосочетаний, соответственно с большей оценкой важности слова  $K(W_i)$ .

**Выделение ключевых слов в найденных концептах.** В концептах высшего уровня могут быть выделены главные смысловые слова (зачастую это термины предметной области) и словосочетания, определяемые в соответствии с результатами ранжирования связанных концептов (по методу главного компонента). Для анализа смысла конструкции тройки «концепт – связь – концепт» в алгоритме формирования онтологии важно привести рассматриваемое словосочетание к одному слову, что позволит представить концепт максимально абстрактно и определить, является ли существенной для будущей онтологии связка с рассматриваемым глаголом. Конструкцию словосочетания в общем случае можно представить как главное слово и зависимые члены. С помощью синтаксического анализатора несложно выделить в словосочетании такие элементы как существительное, числительное или местоимение (при наличии нескольких идентичных элементов в качестве главного принимается слово, приведенное в именительном падеже).

**Определение дочерних концептов и установление связей между ними.** Следующим шагом создания дерева иерархии будущей онтологии является нахождение дочерних концептов по ключевым словам высшего уровня. В качестве дочерних концептов словосочетаний выделяются словосочетания, имеющие наибольшее совпадение слов с ключевыми словами высшего уровня и, соответственно, наибольшую оценку  $M(W_i)$ . На следующем этапе выявляются связи между найденными концептами (по методу главного компонента). Отметим, что для каждого из полученных дочерних концептов могут существовать подчиненные дочерние концепты более низкого уровня (количество уровней обуславливается спецификой предметной области и репрезентативностью исходного корпуса текстов). Алгоритм их выделения в целом аналогичен рассмотренному выше алгоритму поиска концептов высшего уровня, но при этом связан с необходимостью учета значимых связей между концептами смежных уровней.

**Установление связей между концептами онтологии.** Выделим три основных подхода для решения задачи установления связей между концептами проектируемой онтологии:

- поиск слова-претендента на связь в онтологии и последующий подбор концептов, для которых актуальна эта связь (метод 1);
- определение для рассматриваемого концепта списка вероятных слов-претендентов на использование в качестве связи для этого концепта и последующий подбор концепта для установления связи (метод 2);
- нахождение в онтологии двух концептов, которые необходимо связать, и последующий подбор связи для данных концептов (метод 3).

Достоинства первого подхода (метод 1):

- поиск в тексте слов-связей и концептов осуществляется отдельно. Это означает, что концепт и связь не обязательно должны составлять в тексте словосочетание при поиске данной связки в тексте программы автоматического синтеза онтологии;
- возможность варьировать количество учитываемых связок «концепт-связь-концепт» с помощью настраиваемых коэффициентов (уменьшать в случае нахождения большого количества ненужной информации и увеличивать при недостаточном количестве связей в онтологии).

Недостатки первого подхода:

- в общем множестве найденных связей между концептами присутствуют несущественные или несуществующие связи;
- некоторые важные концепты предметной области не имеют связей сформированного множества с другими концептами проектируемой онтологии.

Устранению отмеченных недостатков способствует комбинированное применение второго и третьего подходов (методы 2 и 3).

В работе [6] был предложен метод поиска связей для онтологии, основанный на таком комбинированном подходе, названный методом главного концепта.

Этот метод предполагает необходимость вычисления вероятности применения слова в качестве релевантной связки для рассматриваемого концепта. В качестве слов-связок могут применяться как слова, специфичные для рассматриваемой предметной области, так и достаточно общие слова, которые могут присутствовать в любом тексте. Можно отметить, что слово-связка вероятнее всего будет находиться в тексте между понятиями, которые оно связывает. Вследствие этого целесообразно определить степень специфичности претендента на слово-связку в контексте понятия, которое будет связывать данное слово-связка. В соответствии с предлагаемым алгоритмом считают, что если слово-связка входит в контекст леммы-понятия в рамках рассматриваемого текста, то оно специфично в контексте данного понятия. Введем понятие тройки элементов, используемых для реализации процедуры предварительного отбора наиболее релевантных связок для проектируемой онтологии. К элементам такой тройки отнесем: слово (словосочетание), обозначающее связь между двумя концептами ( $L_1$ ), и собственно два концепта ( $W_1$  и  $W_2$ ), каждый из которых может быть представлен одним словом либо словосочетанием. Таким образом, тройку можно представить в виде: «слово№1, связь, слово№2»:

$$W_1 \leftrightarrow L_1 \leftrightarrow W_2. \quad (5)$$

Отметим, что если концепт представлен словосочетанием, то в тройку вносится главное слово словосочетания.

Выделим четыре возможных варианта представления любой тройки в зависимости от уровня специфичности слова-связки по отношению к понятиям:

$$W_1 \xleftarrow{F_{(w_1,L_1)} \uparrow} L_1 \xleftarrow{F_{(w_2,L_1)} \uparrow} W_2; \quad (6)$$

$$W_1 \xleftarrow{F_{(w_1,L_1)} \uparrow} L_1 \xrightarrow{F_{(w_2,L_1)} \downarrow} W_2; \quad (7)$$

$$W_1 \xrightarrow{F_{(w_1,L_1)} \downarrow} L_1 \xleftarrow{F_{(w_2,L_1)} \uparrow} W_2; \quad (8)$$

$$W_1 \xrightarrow{F_{(w_1,L_1)} \downarrow} L_1 \xrightarrow{F_{(w_2,L_1)} \downarrow} W_2, \quad (9)$$

где символ  $F_{(W_i, L_i)} \uparrow$  означает, что слово-связка  $L_i$  специфично для концепта  $W_i$ , а  $F_{(W_i, L_i)} \downarrow$  означает, что слово-связка  $L_i$  не специфично для концепта  $W_i$ .

На основе статистического анализа текстов рассматриваемой предметной области могут быть определены коэффициенты вероятности  $p_1, p_2, p_3, p_4$  принадлежности определенной тройки к одному из вариантов ее представления: (6), (7), (8) или (9). На основании полученных значений  $p_1, p_2, p_3, p_4$  определим вероятности выбора слова в качестве связки в зависимости от его положения в предложении по отношению к концептам.

При принятии решения о занесении той или иной тройки в проектируемую онтологию, кроме расположения элементов тройки, необходимо учитывать наличие слов между ними и их количество. Очевидно, что целесообразнее вносить в онтологию тройки, элементы которой следуют непосредственно друг за другом, чем тройки, между концептами и связкой которой находятся фрагменты предложения. Назовем расстоянием между элементами тройки количество слов, которые находятся в предложении между двумя любыми элементами тройки. Обозначим через  $N$  расстояние в предложении между двумя концептами  $W_1$  и  $W_2$  рассматриваемой тройки.

Тогда вероятность актуальности рассматриваемой тройки в зависимости от положения ее элементов в предложении можно определить следующим образом:

$$P_{\text{place}} = \frac{k * ( (|m - n| / \min(n, m) + 2) + 1 )}{n + m + 1}, \quad (10)$$

где  $n$  – расстояние от  $L_1$  до  $W_1$ ,  $n = N$ , если между  $L_1$  и  $W_1$  находится  $W_2$ ;  $m$  – расстояние от  $L_1$  до  $W_2$ ,  $m = N$ , если между  $L_1$  и  $W_2$  находится  $W_1$ .

В соответствии с (10), чем больше расстояние между словом-связкой и концептами в тройке, тем меньше вероятность ее актуальности для проектируемой онтологии. Также необходимо отметить, что приведенная формула учитывает приоритет троек, у которых расстояние слова-связки хотя бы с одним из концептов является намного меньше среднего значения такого расстояния для всей совокупности рассматриваемых концептов.

Алгоритм установления связей между концептами онтологии по методу главного концепта можно представить набором следующих действий:

- выбор концепта/понятия и нормализация его до одного слова ( $W_1$ ), для которого следует сформировать тройку в проектируемой онтологии;

- определение множества слов  $M \uparrow (W_1)$ , входящих в контекстное множество данного концепта  $W_1$  (из множества всех слов в предложениях, где присутствует данный концепт с понятием  $M(W_1)$ ), а также множества слов  $M \downarrow (W_1)$ , не входящих в контекстное множество данного концепта;

- определение наиболее вероятного типа связи ( $F_{(W_i, L_i)} \uparrow$  или  $F_{(W_i, L_i)} \downarrow$ ) между концептом  $W_1$  и предполагаемым словом-связкой  $L_i$ ;

- определение множества  $M(L_i)$ , состоящего из претендентов на слова-связки, удовлетворяющих установленному типу связи ( $M(L_i)$ , принимается как  $M \uparrow (W_1)$  или как  $M \downarrow (W_1)$ );

- определение для каждого  $L_i$  (из множества  $M(L_i)$ ) множества слов  $M \uparrow (W_2)$ , входящих в контекстное множество данного слова  $L_i$  (из множества всех слов, входящих в одно предложение с данным словом  $L_i$  и данным словом  $W_1$ , во всех предложениях, где присутствуют  $L_i$  и  $W_1$ ), а также множества слов  $M \downarrow (W_2)$ , не входящих в контекстное множество данного концепта;

- определение множеств  $M_i(W_2)$ , состоящих из претендентов на концепт, связываемый с концептом  $W_1$  при помощи слова-связки  $L_i$ , удовлетворяющих установленному типу связи (для каждого  $L_i$  из множества  $M(L_i)$ );

- определение наиболее вероятного типа связи ( $F_{(W_2, L_1)} \uparrow$  или  $F_{(W_2, L_1)} \downarrow$ ) между будущим словом-связкой  $L_1$  и концептом  $W_2$  (для каждого  $L_1$  из множества  $M(L_1)$ );
- включение в онтологию наиболее вероятной связки из множества  $M(T_1)$  возможных вариантов троек.

На заключительном этапе алгоритма определяется множество  $M(T_1)$  – множество троек, для которых определены типы связей  $F_{(W_1, L_1)}$  и  $F_{(W_2, L_1)}$  ((2), (3), (4) или (5) соответственно). При этом предлагается ранжировать элементы из данного множества в соответствии со значениями вероятностей их выбора в качестве троек, актуальных для проектируемой онтологии.

*Построение онтографа.* Под онтографом, построенным по результатам выполнения предыдущих этапов, понимается двудольный граф, вершинами которого являются концепты высшего уровня и дочерние концепты онтологии предметной области, а дугами – связи между ними. Двудольный граф – это однонаправленный ориентированный граф, в одну вершину которого может входить и выходить несколько дуг.

Для машинного представления данных онтологии и последующего построения онтографа может быть использована реляционная база данных, включающая в себя связанные между собой таблицы концептов и связей между ними (для главного и дочерних уровней). При формировании онтографов на основе таблиц БД могут быть использованы редакторы онтологий (например, редактор Protege [7]). Предложенная модель создания онтологии поддерживает динамическое формирование структуры входящих в состав онтологии предметной области троек «концепт-связка-концепт», позволяя тем самым эффективно реализовать операции поэтапного формирования онтологии и редактирования ее структуры.

### 3. Программная реализация и оценка эффективности предложенного метода

По предложенному методу был разработан программный модуль «Concept-Ont-M2», который может эффективно использоваться для задач анализа электронных текстов и автоматического создания онтологий. Разработанный программный комплекс обеспечивает: автоматическое выделение наиболее значимых слов из корпуса текстов заданной предметной области с последующим их ранжированием по критериям значимости; формирование концептов-словосочетаний для высшего уровня онтологии; формирование троек «концепт-связка-концепт» с последующим выделением дочерних концептов-словосочетаний по методу главного концепта; построение онтографа, отражающего древовидную структуру сформированной онтологической модели. Хранение и коррекция динамически перестраиваемой концептуальной онтологической модели реализуется путем изменения состава и содержания записей, размещаемых в таблицах реляционной базы данных. Программный комплекс обеспечивает как ручной, так и автоматический ввод, редактирование и удаление слов и словосочетаний.

Ниже приведен пример построения по предложенному методу фрагмента онтологии предметной области «Информационные технологии». В качестве входного корпуса текстов использованы электронные тексты авторефератов диссертационных работ по специальности 05.13.06 – информационные технологии. В соответствии с описанным выше подходом выделяем и ранжируем множество концептов-словосочетаний, имеющих наибольшие значения  $K(W_i)$ . Используя формулы (6) – (10), выделяем тройки «концепт – связь – концепт».

Фрагмент ранжированного списка троек «главный концепт – связь – дочерний концепт» имеет вид:

- «модели и информационные технологии администрирования информационного комплекса автоматизированных систем – название – диссертация»;
- «модель актуализации оперативных данных – разработана в – диссертация»;
- «объектная модель типового документа – усовершенствована в – диссертация»;
- «математическая модель – которая описывает – информационный комплекс автоматизированной системы»;
- «математическая модель – позволяет объединить – описание данных»;
- «математическая модель – позволяет объединить – электронные документы»;
- «электронные документы – объединяются по принципу – использование в конкретной функциональной задаче»;

«описание данных – объединяются по принципу – использование в конкретной функциональной задаче»;

«использование в конкретной функциональной задаче – позволяет расширить – функции администрирования»;

«функции администрирования – a part of – информационный комплекс автоматизированной системы»;

«функции администрирования – расширяются за счет – типизации функций»;

«математическая модель – получила дальнейшее развитие в – диссертация»;

«модель запроса – получила дальнейшее развитие в – диссертация»;

«модель запроса – относящийся к – информационный комплекс автоматизированной системы»;

«входные документы – is a – электронные документы»;

«входные документы – воздействуют на состояние – информационный комплекс»;

«модель актуализации оперативных данных – is a – модель актуализации оперативных данных информационного комплекса»;

«информационный комплекс – is a – информационный комплекс автоматизированной системы»;

«модель актуализации оперативных данных – позволяет автоматизировать – функция обработки документов»;

«модель актуализации оперативных данных – позволяет получить – количественная оценка фактографических данных»;

«объектная модель типового документа – представляет – электронные документы»;

«электронные документы – представляются как – иерархия модифицированных фреймов».

После ранжирования списка троек «концепт – связь – концепт» для каждого главного концепта строится корневое дерево связанных с ним концептов, в котором корнем является главный концепт, вершинами на нижних уровнях – связанные с ним концепты, а дугами – связи между концептами. Связи между концептами могут быть традиционными (is-a, a-part-of), а могут представлять собой глагольные группы, определенные при помощи метода главного концепта.

На рис. 1 представлен фрагмент построения корневых деревьев, корнями которых являются главные концепты, вершинами – связанные концепты, а дугами – связи между ними (наименования концептов и связей приведены в сокращенном виде – полностью открываются с помощью кликов). Если некоторые концепты присутствуют в нескольких деревьях, то при объединении деревьев повторяющиеся вершины отождествляются.

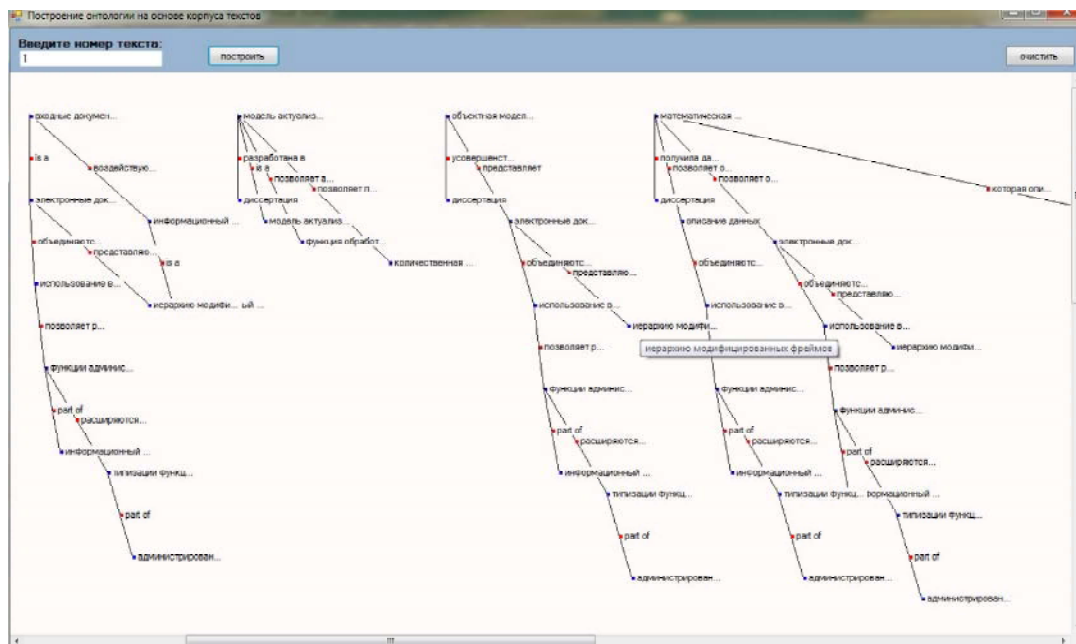


Рис.1

Для построения онтологии документа все построенные деревья объединяются в один онтограф (который может и не быть деревом). На рис.2 представлен фрагмент онтографа, построенного с учетом связей по методу главного концепта для одного текста. В построенном онтографе отождествлены идентичные вершины, соответствующие совпадающим концептам.

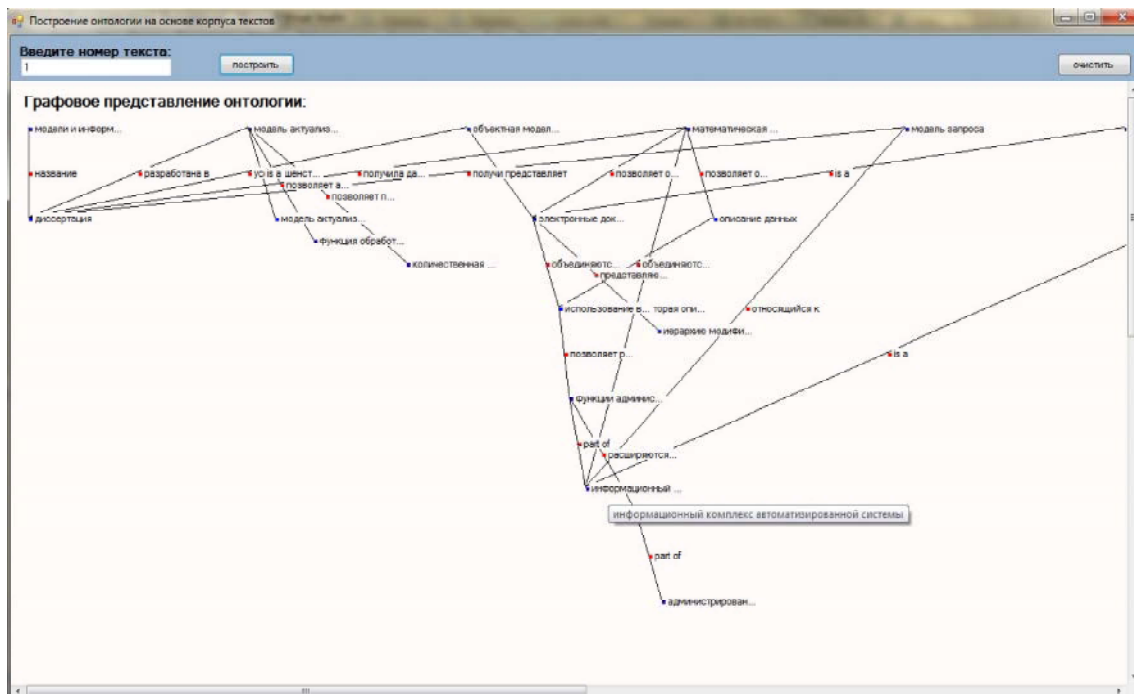


Рис. 2

Проведенные экспериментальные исследования показали, что онтологические модели, построенные с применением предложенного подхода (в частности, метода главного концепта) являются более представительными, чем модели, основанные на поиске отдельных связей между концептами. Оценка эффективности проводилась по двум параметрам:  $R$  – точность поиска связей (отношение правильно найденных связей к общему количеству найденных связей);  $P$  – полнота поиска связей (отношение правильно найденных связей к общему количеству связей, выявленных экспертом). Результаты экспериментальных исследований (для корпуса текстов из электронной библиотеки авторефератов по технической тематике): для метода поиска отдельных связей и метода главного концепта значения  $R$  составляют 59 и 83% соответственно; значения  $P$  – 79 и 86% соответственно.

#### 4. Выводы и перспективы дальнейших исследований

Проведенные исследования позволяют сделать вывод, что важным этапом автоматического построения онтологий является выявление релевантных связей между концептами-словосочетаниями с последующим формированием троек «концепт – связь – концепт». Модификация и программная реализация метода нахождения таких связей с учетом расположения элементов «концепт-связка» в тексте позволили повысить эффективность автоматического построения онтологических моделей в виде онтографа. Вспомогательным этапом такого построения есть формирование корневого дерева, в котором корнем является главный концепт, вершинами на нижних уровнях – связанные с ним концепты, а дугами – связи между концептами. Если некоторые концепты присутствуют в нескольких деревьях, то при объединении деревьев повторяющиеся вершины отождествляются.

Научная новизна предложенного метода состоит в реализации оригинальной процедуры автоматического построения древовидной структуры концептов формируемой онтологии в рамках анализируемого корпуса текстов. При проведении дальнейших исследований целесообразно усовершенствовать предложенный метод, дополнив его анализом более сложных типов связей в онтологической модели и учетом дополнительных атрибутов формируемых онтологий.

Список литературы: **1.** Палагин А.В. Онтологические методы и средства обработки предметных знаний: Монография / А.В. Палагин, С.Л. Кривый, Н.Г. Петренко. Луганск: изд-во ВНУ им. В. Даля, 2012. 324 с. **2.** Норенков И.П. Интеллектуальные технологии на базе онтологий / И.П. Норенков // Информационные технологии. 2010. №1. С.17-23. **3.** Антонов И.В. Методы анализа данных в задачах автоматизации построения онтологии предметной области / И.В. Антонов, М.В. Воронов // Дистанционное и виртуальное обучение. 2011. N 8. С. 19-35. **4.** Палагин О.В. Розбудова абстрактної моделі мовно-онтологічної інформаційної системи / О.В. Палагин, М.Г. Петренко // Математичні машини і системи. 2007. №1. С. 42–50. **5.** Чалая Л. Э. Меры важности концептов в семантической сети онтологической базы знаний [Текст] / Л. Э. Чалая, Ю. Ю. Шевякова, А. Ю. Шафроненко // Матеріали другої міжнар. наук.-техн. конф. «Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління». Киев : КДАВТ, 2011. С. 51. **6.** Чалая Л. Э. Определение значимости связей между концептами при автоматическом синтезе онтологий / Л. Э. Чалая, А. В. Чижевский // International Scientific Journal «Acta Universitatis Pontica Euxinus» (Special number). Varna. 2014. P. 480–484.

*Поступила в редколлегию 17.12.2015*

**Чалая Лариса Эрнестовна**, канд. техн. наук, доцент кафедры искусственного интеллекта ХНУРЭ. Адрес: Украина, 61166, Харьков, пр. Науки, 14, тел. 057-7021337, e-mail: larysa.chala@nure.ua.

**Чижевский Антон Валериевич**, аспирант кафедры искусственного интеллекта ХНУРЭ. Адрес: Украина, 61166, Харьков, пр. Науки, 14, тел. 057-7021337, e-mail: chij7@mail.ru

## **ЗАСТОСУВАННЯ ГЕНЕТИЧНИХ АЛГОРИТМІВ У ЗАДАЧІ ПРО УКЛАДАННЯ РАНЦЯ**

---

Розглядається застосування генетичних алгоритмів у криптології. Описуються основні поняття генетичного алгоритму, представляється його цикл у вигляді блок-схеми та приведена послідовність етапів роботи алгоритму. Формулюється задача про укладання ранця та загальний алгоритм її розв'язання. На прикладі задачі про укладання ранця розглядається можливість застосування генетичних алгоритмів при шифруванні. Показується, що застосування генетичних алгоритмів – один із найкращих методів для пошуку «прийнятних» рішень.

### **1. Вступ**

Ідея генетичних алгоритмів запозичена з живої природи. Вона полягає в машинній організації еволюційного процесу створення, модифікації і відбору кращих розв'язків. У загальному значенні *генетичні алгоритми* – це тип алгоритмів, інспірованих механізмами еволюції живої природи, які застосовуються до задач глобальної оптимізації, для комбінування шаблонів з правил індукції, що були відкриті до цього, навчання нейромереж, пошуку зразків у даних, відкриття шаблонів у тексті тощо. Методологічна основа генетичних алгоритмів ґрунтується на гіпотезі селекції, яка в загальному виді може бути сформульована так: чим вища пристосованість особини, тим вища ймовірність того, що в потомстві, отриманому з її участю, ознаки, які визначають пристосованість, будуть виражені ще сильніше. Генетичні алгоритми застосовуються в багатьох задачах, в тому числі і в задачах криптології.

Шифрування інформації в наш час стало чи не основним методом її захисту. Доступність обчислювальної техніки й стрімкий прогрес у її розвитку привели до вдосконалювання давно відомих шифрів і застосування в масовому масштабі нових високонадійних схем шифрування. Однак цей прогрес має й інший бік: збільшені можливості обчислювальної техніки успішно застосовуються не тільки для шифрування, але й для дешифрування тих шифрів, які ще зовсім недавно гарантували повний захист інформації.

Основними завданнями криптології є розробка надійних схем шифрування (завдання криптографії) і знаходження ефективних методів дешифрування існуючих схем (завдання криптоаналізу). Криптографічний спосіб захисту інформації передбачає таке її перетворення, при якому вона стає доступною для прочитання лише власникові секретного ключа. Надійність цього способу захисту визначається стійкістю використовуваної схеми шифрування до криптоаналізу. При криптоаналізі конкретного шифру передбачається, що сама схема шифрування відома, а невідомим є тільки секретний ключ і/або його довжина. Іншими словами, завдання розкриття шифру полягає в знаходженні єдиного справжнього секретного ключа серед безлічі всіх можливих ключів, тобто є завданням пошуку. При цьому простір пошуку великий, а критерій «якості» знайденого розв'язку, як правило, не піддається строгій формалізації. В наш час у криптоаналізі успішним є застосування генетичних алгоритмів.

*Метою* дослідження є розробка альтернативного способу розв'язання задачі про укладання ранця.

### **2. Основні поняття генетичного алгоритму**

Застосування генетичних алгоритмів вперше запропоновано Джоном Холландом. Ці алгоритми являють собою модифікацію так званого «еволюційного програмування». Ідея Холланда полягала в тому, щоб розробити алгоритми на основі «спрямованого» випадкового пошуку або на основі механізмів природного відбору, відомого з біології. На етапі ініціалізації цієї процедури створюється популяція можливих розв'язків. У результаті з цієї популяції виводиться нове покоління розв'язків, яке, у свою чергу, служить «вихідним

матеріалом» для чергового покоління [1]. Представимо цикл генетичного алгоритму у вигляді блок-схеми (рис. 1), який включає стадії відбору, схрещування й мутації. Аналізуючи представлену блок-схему, відмітимо, що кращі представники покоління відбираються для відтворення популяції. Таким чином, можна припустити, що кожне нове покоління повинне містити кращі розв'язки, ніж попереднє.

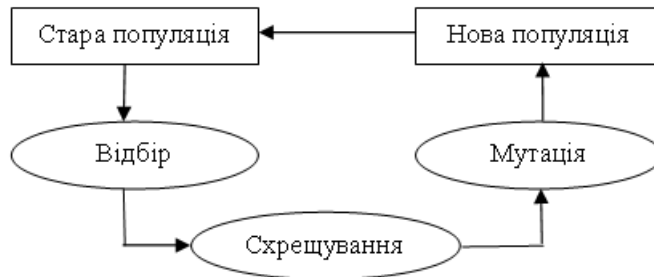


Рис. 1. Блок-схема еволюції популяції

Розглянемо принцип роботи блок-схеми, представлені на рис. 1.

Стара популяція складається з набору бінарних рядків. Кожний бінарний рядок представляє розв'язок проблеми й називається хромосомою. Першою стадією генетичного алгоритму є відбір. У процесі відбору визначаються рядки, які будуть використовуватися при створенні нової популяції (нового покоління). «Батьки» вибираються довільно, однак «кращі особини» популяції мають більший шанс виявитися обраними. Таким чином, алгоритм «просувається» в перспективному напрямку пошуку.

Наступна стадія – схрещування, яке полягає в тому, що для «батьків» (пари відібраних рядків – пари хромосом) довжини  $r$  ( $r$  – кількість бітів у хромосомі) кожна наступна пара «нащадків» вибирається довільним чином. «Нашадки» формуються з частини бітів  $s$ , де  $s \in \{1, \dots, r\}$ , і бітів  $s'$  та  $s''$ , де  $s'$  і  $s''$  – це частина бітів від  $s+1$  до  $r$ , якими обмінюються «батьки» (рис. 2).

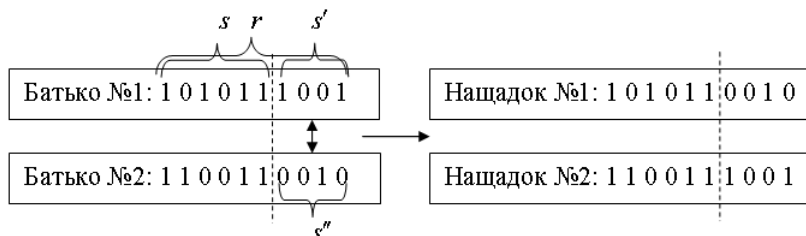


Рис. 2. Передача генетичної інформації від «батьків» до «нащадків»

Заключна стадія – мутація. При ініціалізації алгоритму встановлюється ймовірність мутації, якої зазнають новоутворені хромосоми. Для прикладу, наведеного на рис. 2, розглянемо таку мутацію: «нащадок №1» отримав мутацію 8-го біта, а «нащадок №2» отримав мутацію 2-го і 10-го біта (рис. 3).

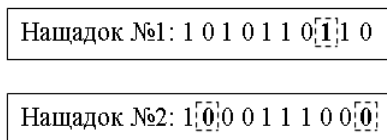


Рис. 3. Процес мутації серед «нащадків»

Схема еволюції популяції (див. рис. 1) буде виконана як мінімум один раз. Стадії еволюції повторюються до досягнення умови виходу із циклу (такою умовою може бути, наприклад, перевищення максимальної кількості популяцій).

### 3. Шифрування за допомогою задачі про укладання ранця

Один із перших шифрів на основі задачі про укладання ранця був запропонований Мерклі й Хеллманом в 1978 [2]. Це була одна з перших спроб створення системи шифрування з

відкритим ключем. Незважаючи на те, що проблема укладання ранця відноситься до класу NP-повних [3], було показано, що більшість версій алгоритму є нестійкими. В 1983 р. Брікел запропонував спосіб злому криптосистеми на основі ранця низької щільності [4]. Рік по тому Шамір розробив поліноміальний алгоритм для атаки на вихідну «рюкзачну» криптосистему [5]. Після цього було запропоновано безліч інших систем на основі алгоритму укладання ранця: кілька послідовних рюкзаків, рюкзаки Грем–Шаміра [6]. Для всіх цих систем були розроблені методи розкриття. У статті [7] пропонується ще один метод криптоаналізу шифрів на основі алгоритму укладання ранця; відмінною рисою такого підходу є його універсальність, тобто можливість застосування до будь-якої версії «рюкзачної» криптосистеми, а також простота роботи. Метод базується на використанні генетичних алгоритмів.

Розглянемо формулювання задачі про укладання ранця. Задана множина предметів різної ваги; запитується, чи можна покласти деякі із цих предметів у ранець так, щоб його вага стала дорівнювати певному значенню? Більш формально задача формулюється так: даний набір значень  $M_1, M_2, \dots, M_n$  і сумарне значення  $S$ , потрібно обчислити значення  $b_i$  ( $i = \overline{0, n}$ ) такі, що:  $S = b_1 M_1 + b_2 M_2 + \dots + b_n M_n$ .

Тут  $b_i$  може бути або нулем, або одиницею. Значення  $b_i = 1$  означає, що  $i$ -й предмет кладуть у рюкзак, а  $b_i = 0$  – не кладуть. Звідси впливає уявлення про вміст рюкзака у вигляді хромосом, біти яких відповідають значенням  $b_i$ . Функція вибору «кращих хромосом» оцінює близькість ваги конкретного рюкзака до заданого числа. Значення функції розташовуються в діапазоні  $[0, 1]$ , де 1 означає точний збіг із шуканою вагою. Якщо вага одного рюкзака перевищує цільове значення  $S$  на деяке число  $x$ , а вага іншого, навпаки, менша за потрібну на те ж число  $x$ , то «кращим» вважається останній рюкзак. Опишемо цю задачу більш формально:

1. Потрібно обчислити максимальну розбіжність, яка може виникнути між довільною хромосомою й цільовим значенням  $S$ :  $\Delta_{\max} = \max(S, \tilde{S} - S)$ , де  $\tilde{S}$  – сума всіх компонентів, які можна використовувати при укладанні рюкзака.

2. Обчислити вагу рюкзака, відповідного до поточної хромосоми, і позначити  $S'$ .

3. Якщо  $S' \leq S$ , то «якість» хромосоми оцінюється значенням:  $\alpha = 1 - \sqrt{\frac{|S' - S|}{S}}$ .

4. Якщо  $S' > S$ , то «якість» хромосоми оцінюється значенням:  $\alpha = 1 - \sqrt{\frac{|S' - S|}{\Delta_{\max}}}$ .

Загальний алгоритм задачі про укладання ранця можна представити так:

1. Створюється випадкова популяція двійкових хромосом.
2. Для кожної хромосоми обчислюється значення  $\alpha$  (функція оцінки).
3. На основі отриманих коефіцієнтів відбувається природний відбір.
4. До обраних на 3-му етапі особин застосовується схрещування.
5. Нащадки зазнають мутації.
6. Нова популяція аналізується, виділяються «кращі хромосоми».

Процес перерветься, коли кількість поколінь перевищить певне задане число. «Кращі хромосоми» нового покоління будуть використані для «злому» шифру [8, 9].

Розглянемо приклад шифрування за допомогою задачі про ранець.

*Рюкзачний вектор*  $A = (a_1, a_2, \dots, a_n)$  – впорядкований набір із  $n$  предметів. Повідомлення шифрується як розв'язок набору задач про ранець. Для шифрування відкритого тексту у двійковому представленні його розбивають на блоки довжини  $n$  (наприклад, (1 1 1 0 0) відповідає 5-ти предметам у рюкзаку). Вважається, що одиниця вказує на наявність предмета в рюкзаку, а нуль – на його відсутність. Якщо для заданого рюкзачного вектора  $A = (3, 4, 5, 6, 7, 8)$  довжини  $n = 6$  за даним алгоритмом шукати шифротекст, то усі криптосистеми не перевищуватимуть 33 – сумарну вагу всіх предметів у рюкзачному векторі (таблиця).

Шифрування тексту у двійковому представленні

Відкритий текст	Речі в рюкзаку	Шифротекст
0 0 0 0 0 0	0·3 + 0·4 + 0·5 + 0·6 + 0·7 + 0·8	0
0 0 0 0 0 1	0·3 + 0·4 + 0·5 + 0·6 + 0·7 + 1·8	8
0 0 0 0 1 0	0·3 + 0·4 + 0·5 + 0·6 + 1·7 + 0·8	7
0 0 0 0 1 1	0·3 + 0·4 + 0·5 + 0·6 + 0·7 + 0·8	15
0 0 0 1 0 0	0·3 + 0·4 + 0·5 + 0·6 + 1·7 + 1·8	6
0 0 0 1 0 1	0·3 + 0·4 + 0·5 + 1·6 + 0·7 + 1·8	14
0 0 0 1 1 0	0·3 + 0·4 + 0·5 + 1·6 + 1·7 + 0·8	13
0 0 0 1 1 1	0·3 + 0·4 + 0·5 + 1·6 + 1·7 + 1·8	21
0 0 1 0 0 0	0·3 + 0·4 + 1·5 + 0·6 + 0·7 + 0·8	5
0 0 1 0 0 1	0·3 + 0·4 + 1·5 + 0·6 + 0·7 + 1·8	13
0 0 1 0 1 0	0·3 + 0·4 + 1·5 + 0·6 + 1·7 + 0·8	12
0 0 1 0 1 1	0·3 + 0·4 + 1·5 + 0·6 + 1·7 + 1·8	20
0 0 1 1 0 0	0·3 + 0·4 + 1·5 + 1·6 + 0·7 + 0·8	11
0 0 1 1 0 1	0·3 + 0·4 + 1·5 + 1·6 + 0·7 + 1·8	19
0 0 1 1 1 0	0·3 + 0·4 + 1·5 + 1·6 + 1·7 + 0·8	18
0 0 1 1 1 1	0·3 + 0·4 + 1·5 + 1·6 + 1·7 + 1·8	26
0 1 0 0 0 0	0·3 + 1·4 + 0·5 + 0·6 + 0·7 + 0·8	4
0 1 0 0 0 1	0·3 + 1·4 + 0·5 + 0·6 + 0·7 + 1·8	12
0 1 0 0 1 0	0·3 + 1·4 + 0·5 + 0·6 + 1·7 + 0·8	11
0 1 0 0 1 1	0·3 + 1·4 + 0·5 + 0·6 + 1·7 + 1·8	19
0 1 0 1 0 0	0·3 + 1·4 + 0·5 + 1·6 + 0·7 + 0·8	10
0 1 0 1 0 1	0·3 + 1·4 + 0·5 + 1·6 + 0·7 + 1·8	18
0 1 0 1 1 0	0·3 + 1·4 + 0·5 + 1·6 + 1·7 + 0·8	17
0 1 0 1 1 1	0·3 + 1·4 + 0·5 + 1·6 + 1·7 + 1·8	25
0 1 1 0 0 0	0·3 + 1·4 + 1·5 + 0·6 + 0·7 + 0·8	9
0 1 1 0 0 1	0·3 + 1·4 + 1·5 + 0·6 + 0·7 + 1·8	17
0 1 1 0 1 0	0·3 + 1·4 + 1·5 + 0·6 + 1·7 + 0·8	16
0 1 1 0 1 1	0·3 + 1·4 + 1·5 + 0·6 + 1·7 + 1·8	24
0 1 1 1 0 0	0·3 + 1·4 + 1·5 + 1·6 + 0·7 + 0·8	15
0 1 1 1 0 1	0·3 + 1·4 + 1·5 + 1·6 + 0·7 + 1·8	23
0 1 1 1 1 0	0·3 + 1·4 + 1·5 + 1·6 + 1·7 + 0·8	22
0 1 1 1 1 1	0·3 + 1·4 + 1·5 + 1·6 + 1·7 + 1·8	30
1 0 0 0 0 0	1·3 + 0·4 + 0·5 + 0·6 + 0·7 + 0·8	3
1 0 0 0 0 1	1·3 + 0·4 + 0·5 + 0·6 + 0·7 + 1·8	11
1 0 0 0 1 0	1·3 + 0·4 + 0·5 + 0·6 + 1·7 + 0·8	10
1 0 0 0 1 1	1·3 + 0·4 + 0·5 + 0·6 + 1·7 + 1·8	18
1 0 0 1 0 0	1·3 + 0·4 + 0·5 + 1·6 + 0·7 + 0·8	9
1 0 0 1 0 1	1·3 + 0·4 + 0·5 + 1·6 + 0·7 + 1·8	17
1 0 0 1 1 0	1·3 + 0·4 + 0·5 + 1·6 + 1·7 + 0·8	16
1 0 0 1 1 1	1·3 + 0·4 + 0·5 + 1·6 + 1·7 + 1·8	24
1 0 1 0 0 0	1·3 + 0·4 + 1·5 + 0·6 + 0·7 + 0·8	8
1 0 1 0 0 1	1·3 + 0·4 + 1·5 + 0·6 + 0·7 + 1·8	16
1 0 1 0 1 0	1·3 + 0·4 + 1·5 + 0·6 + 1·7 + 0·8	15
1 0 1 0 1 1	1·3 + 0·4 + 1·5 + 0·6 + 1·7 + 1·8	23

Для кожного вихідного тексту існує єдиний унікальний криптотекст.

Задача про ранець відноситься до класу NP-повних задач [3], для неї немає поліноміального алгоритму, що вирішує її за розумний час. Тому при розв'язуванні задачі про ранець завжди потрібно вибирати між точними алгоритмами, які не застосовні для «великих» рюкзаків, і наближеними, які працюють швидко, але не забезпечують оптимального розв'язку задачі. Природно, створення швидкого й достатньо точного алгоритму становить великий інтерес.

#### 4. Висновки

Наведені вище основні особливості застосування генетичних алгоритмів, які імітують процеси еволюції живої природи, свідчать про ефективність їх застосування для розв'язування задач криптології. «Сила» генетичного алгоритму полягає в його здатності оперувати одночасно багатьма параметрами, які використовуються в сотнях прикладних програм. У деяких випадках потрібно знайти параметри, при яких досягається точне значення результату. В інших випадках точний оптимум не потрібний – рішенням може бути будь-яке значення, краще за певну задану величину. У цьому випадку генетичні алгоритми – найкращий метод для пошуку «прийнятних» рішень.

**Список літератури:** 1. *Holland J.H.* Adaptation in Natural and Artificial Systems / J.H. Holland. Ann Arbor, MI: The University of Michigan Press, 1975. 2nd edn. Boston, MA: MIT Press, 1992. 2. *Merkle R.C.* Hiding Information and Signatures in Trapdoor Knapsacks / R.C. Merkle, M.E. Hellman // IEEE transactions on Information Theory. Sep 1978. V. 24. N. 5. P. 525–530. 3. *Ахо А.* Построение и анализ вычислительных алгоритмов / А. Ахо, Д. Хопкрофт, Д. Ульман. М.: Мир, 1979. С. 404–446. 4. *Brickell E.* Solving Low Density Knapsacks / E. Brickell // Advances in Cryptology: Proceedings of crypto. New York: Plenum Press, 1984. P. 25–37. 5. *Shamir A.* A Polynomial – Time Algorithm for Breaking the Basic Merkle – Hellman Cryptosystem / A. Shamir // Proceedings of the 23rd IEEE Symposium on the Foundations of Computer Science. 1982. P. 145–152. 6. *Schneier B.* Applied Cryptography Second Edition: protocols, algorithms and source code in C. John Wiley & Sons Inc., 1996. (Русский перевод: Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. М.: ТРИУМФ, 2002). 7. *Cryptanalysis of knapsack ciphers using genetic algorithms / R. Spillman // Cryptologia. 1993. V. 17. N. 4. P. 367–377.* 8. *Matthews R.* The use of genetic algorithms in cryptanalysts / R. Matthews // Cryptologia. 1993. V. 17. N. 2. P. 187–201. 9. *Use of a Genetic Algorithm in the Cryptanalysis of Simple Substitution Ciphers / [R. Spillman, M. Janssen, B. Nelson, M. Kepner] // Cryptologia. 1993. V. 17. N. 1. P. 31–44.*

Надійшла до редколегії 23.12.2015

**Кожухівський Андрій Дмитрович**, д-р техн. наук, професор кафедри інформатики та інформаційної безпеки Черкаського державного технологічного університету. Наукові інтереси: аналіз і моделювання складних систем. Адреса: Україна, 18006, Черкаси, бульвар Шевченка, 460, тел. 0472730217. E-mail: andrejdc@mail.ru

**Намофілова Ольга Олексіївна**, аспірантка кафедри інформатики та інформаційної безпеки Черкаського державного технологічного університету. Наукові інтереси: математичне моделювання, генетичні алгоритми. Адреса: Україна, 18006, Черкаси, бульвар Шевченка, 460, тел. 0472730217. E-mail: olga\_namofilova@rambler.ru

## **КОМПЬЮТИНГОВЫЕ МОДЕЛИ ОБЛАЧНЫХ СЕРВИСОВ**

---

Предлагается киберкультура микро-макро-космо-компьютинга, которая формулирует, объясняет и прогнозирует современные технологии мониторинга и управления процессами и явлениями в физическом, виртуальном и космологическом пространстве. Представляются вербальные и структурные определения главных типов компьютеринга, основанные на современных трендах эволюционного развития киберэкосистемы планеты. Формулируется универсальная модель МАТ-компьютинга: <Memory, Address, Transactions>, которая использует три компонента для создания вычислительной структуры в технологически приемлемой материальной среде. Показывается информационно-квантовое направление экспансии человека в космическое пространство, а также возможность аналогичного проникновения неземных биотехнических объектов в экосистему нашей планеты. Предлагается модель компьютеринга, которая задает квазиоптимальные структуры мониторинга и управления масштабируемыми процессами различной природы: техническими, биологическими, социальными, виртуальными и космологическими.

### **1. Введение**

Рассматривается автоматная модель устойчивого развития человечества, заданная в метрике зеленой концепции сохранения планеты и повышения качества жизни человечества [Vyacheslav Kharchenko and Oleg Illiashenko. *Concepts of Green IT Engineering: Taxonomy, Principles and Implementation*. Springer-Verlag Berlin Heidelberg 2016]. Для формирования модели следует определить трендовые аксиомы (The 10 Commitments of Greening), которым необходимо следовать для устойчивого развития человечества:

1) Использование автоматной модели компьютеринга для мониторинга и управления всеми процессами и явлениями, как самой надежной и простой для понимания и исполнения. Модель имеет минимальное число компонентов и сигналов при реализации замкнутой детерминированной компьютеринговой системы для достижения поставленной цели. Жизненность автоматной модели подтверждается не только созданием индустрии компьютеров, но и адекватным компьютеринговым описанием процессов и явлений в физике, биологии, медицине, технике, социологии, космологии.

2) Механизмы исполнения и управления при реализации модели компьютеринга не должны пересекаться по составу компонентов. Условие регламентирует строгое разделение функциональных обязанностей между механизмами управления и исполнения, что также означает невмешательство одного механизма в дела другого. Нарушение данного пункта является типичной системной ошибкой, порождающей коррупцию, и свойственной управлению социальными группами.

3) Применение концепции Бога – все видит и воздает по заслугам – для мониторинга и управления индивидуумом, социальными группами и человечеством в рамках создания облачных компьютеринговых сервисов. Устойчиво исключать человека, как самое ненадежное звено, из мониторинга и управления киберфизическими, биологическими, техническими, технологическими и социальными процессами и явлениями. Условия позволяют оптимально управлять каждым человеком, социальными группами и государствами, что исключает все негативные последствия от вмешательства человека, включая войны, социальные конфликты, коррупцию и несправедливость. Экономический эффект от реализации концепции измеряется десятками триллионов долларов.

4) Автоматный детерминизм и предсказуемость реакции компьютеринга на инициирующие или актуаторные воздействия. Исключение вероятности из событий и процессов, как “фигового листка на голом теле нашего невежества”, по определению Эйнштейна. Условия ориентированы на детерминизм компьютеринговых сервисов, максимально исключая непредсказуемость вероятностных методов, что существенно уменьшает накладные расходы на дублирование и резервирование системных компонентов и процессов.

5) Тотальное оцифровывание всех пространственных, киберфизических, биологических, социальных процессов, явлений и транспорта, обеспечивающих точное управление на основе цифрового мониторинга и позиционирования. Условия позволяют в глобальном масштабе решить проблему надежного безаварийного, беспилотного управления всеми видами транспорта, уменьшить расходы на изготовление автомобильных номеров (5 млрд долларов), инфраструктуру дорожных знаков и светофоров (500 млрд долларов) за счет создания виртуальной e-инфраструктуры управления транспортом. В настоящее время на дорогах всех стран погибает в год 1,2 миллиона человек. Цифровой мониторинг природных явлений, ураганов, тайфунов, землетрясений, цунами, потепления на основе внедрения сети сенсоров обеспечит актуаторное управление климатическими и геопатогенными катастрофами. Внедрение технологии SlingShot в качестве специализированного решения проблемы чистой воды, от недостатка которой ежегодно умирает 2 миллиона жителей планеты, путем точного мониторинга загрязняющих компонентов и их последующего устранения адекватными актуаторными воздействиями.

6) Метрическое оценивание всех процессов и явлений, формирующих интегральный критерий “время–деньги–качество” в целях морально-материального стимулирования социально и экологически значимых проектов. Метрическое оценивание и последующее адекватное морально-материальное стимулирование членов социальных групп и коллективов, оказывающих существенное влияние на позитивные процессы повышения качества жизни людей и экосистемы планеты. Реализация глобального рыночного тренда e-infrastructure от программы EU Horizon 2020 применительно ко всем земным процессам и явлениям, включая: health care e-infrastructure, smart home e-infrastructure, smart university e-infrastructure, smart city e-infrastructure, social management and government e-infrastructure, traffic control e-infrastructure, internet driven e-infrastructure for diagnosis and repair of technics, computers.

7) Полное исключение вторичных признаков идентификации человека в киберфизическом пространстве на основе повсеместного внедрения e-интерфейсов для ввода первичных аутентификаторов (отпечатки пальцев, радужная оболочка глаза, ДНК). Условия позволяют убрать из обращения миллиарды бумажных паспортов и электронных карточек, которые можно подделать, потерять, что дает возможность сохранить леса и сэкономить до 10 миллиардов долларов на изготовление документов.

8) Создание глобальной e-инфраструктуры надежного, защищенного доступа к киберфизическому пространству планеты и разработка соответствующего законодательства, обеспечивающих легитимное online исполнение функциональных обязанностей, не привязанных к рабочему месту. Условия позволяют в масштабах планеты на 20 процентов уменьшить транспортный трафик, потребление бензина и вредных выбросов, благодаря выполнению функциональных обязанностей работниками по месту жительства.

9) Внедрение электронного гражданства для всех жителей планеты и исключение бумажных носителей информации из всех сфер человеческой деятельности. Условия позволяют сохранить не менее 20 процентов леса от вырубki для изготовления бумаги, что повысит содержание кислорода на планете и улучшит ее экологию. Электронное гражданство сократит миграцию граждан и сделает все страны привлекательными, из-за реальной опасности, для руководителей отсталых государств, электронного ухода всех граждан вместе с их налогами.

10) Облачный и big data компьютеринг на основе планетарной сети дата центров позволяет решить проблему защиты информации, сервисов, персональных данных, убрать миллионы настольных компьютеров, локальных серверов и перейти к использованию экономичных энергосберегающих планшетов доступа к сервисам и персональным кабинетам. Brain-Computer интерфейсы дают возможность устранить многочисленные устройства ввода данных и перейти к непосредственному образно-импульсному общению человеческого мозга с компьютеринговыми терминалами. Сканирование мозга на основе внешних или встроенных сенсоров позволяет предотвращать преступные или нелегитимные действия человека, что существенно повлияет на эффективность работы полиции и специальных служб.

Выполнение перечисленных аксиом позволит переместить все компоненты механизма управления земными процессами и явлениями в облака компьютерных сервисов, что освободит существенную часть (20%) человечества от выполнения несвойственных ему управленческих непроизводительных процессов. Отдать управление облачному компьютеру означает существенно снизить накладные расходы и открыть путь к созданию зеленой планеты (качество жизни + чистая экология).

Стоимость реализации перечисленных аксиом – порядка 50 миллиардов долларов, экономия от их внедрения – не менее 50 триллионов долларов плюс качество жизни каждого человека и реинкорнация экологии зеленой планеты.

Все технические проблемы для реализации десяти трендов уже решены в той или иной степени. Главным препятствием на пути создания зеленой планеты и счастливой жизни является невысокий уровень киберкультуры человека. Поэтому проблема достижимости поставленных зеленых целей жестко связана с воспитанием доверия у людей к надежности, справедливости и неподкупности компьютерных сервисов мониторинга и управления.

Устойчивое развитие зеленого компьютеризма представлено в метрике пространства и времени тремя историческими периодами или фазами (рис. 1).

1) Отображение (мониторинг) физических процессов и явлений, представленное сингулярным (Single Computing), сетевым (Network Computing) и глобальным компьютеризмом (Global Computing – Internet). Здесь также фигурируют понятия Desktop, Servers, Data Base.

2) Управление физическими процессами и явлениями на основе e-infrastructure, цифрового мониторинга, представленное Cloud Computing, Cyber Physical Networks, Internet of Things. Цикл настоящего времени, где основными участниками являются Gadget, Laptop, Data Centers, Big Data.

3) Созидание интеллектуальных киберфизических процессов и явлений под управлением кибермозга планеты, представленное Brain Computing, Robotic Networks, Internet of Nature (World). Здесь ожидается появление Massive Quantum-Atomic Computing, Brain-Computer Interface, Atomic Data Center Networks, Smart Big Data Networks.

Здесь же, на рис. 1, представлены векторы развития зеленого компьютеризма, которые оформлены в следующие фазовые трансформации по трем историческим периодам: 1) Single Computing – Cloud Computing – Brain Computing. 2) Network Computing – Cyberphysical Computing – Network Robotic Computing. 3) Internet Computing – Internet of Things – Internet of Nature. 4) Data based Computing – Big Data Computing – Smart Data Computing.



Рис. 1. Computing in sustainable development

Космологическая модель Вселенной представлена жестко связанными между собой взаимодействующими компонентами: Пространство и Время, Материя и Энергия. Данную структуру базовых понятий можно применить и к процессу устойчивого развития компьютеринга. Для этого необходимо выделить следующие компоненты: 1) Масса материи ( $m$ ) для реализации примитива-транзистора или компьютера. 2) Энергия ( $E$ ) для выполнения элементарной операции или функционирования компьютера. 3) Быстродействие ( $t$ ) как величина, обратная времени ( $t=1/T$ ) выполнения одной элементарной операции, или производительность компьютера ( $t=I/T$ , количество инструкций в секунду). 4) Пространство ( $S$ ) (киберфизическое), обслуживаемое компьютерингом путем предоставления пользователям сервисов мониторинга, управления, созидания.

Используя введенные компоненты в качестве аргументов, можно определить на исторически незначительном промежутке времени (1970-2020) два достаточно парадоксальных и очень оптимистически зеленых закона развития компьютеринга (рис. 2): 1) Связь между энергией и временем. Повышение быстродействия компьютера связано с уменьшением энергопотребления. 2) Связь между материей и пространством. Уменьшение массы компьютера связано с расширением обслуживаемого пространства. Оба закона верны и в обратном прочтении.

Если выполнить суперпозицию двух законов, то можно получить интегральный закон компьютеринга: увеличение быстродействия и пространства компьютеринга жестко связано с уменьшением его энергопотребления и массы. Если условно определить интервал изменения относительных значений всех четырех параметров между 0 и 1, то закон можно записать в следующем виде:  $t+E=1$ ,  $S+m=1$ . Это означает, что аддитивная оценка быстродействия (пространства) и энергопотребления (массы) компьютеринга на полувековом отрезке времени есть величина постоянная.

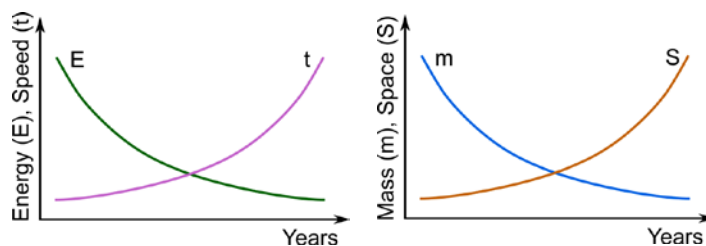


Рис. 2. Computing components interaction

Наиболее значимые рыночно-ориентированные открытия и инновации ученые делают путем суперпозиции междисциплинарных исследований, а также имплементации достижений одной технологической культуры в смежные отрасли знаний. Модели компьютеринга все шире применяются для мониторинга и управления процессами и явлениями во всех областях деятельности человека и природы [1-4]. Интеграция природной и биосоциальной культуры с кибертехнологическими решениями мониторинга и управления приводит к оригинальным системным (киберфизическим, биоинформационным) научным результатам и инновациям в традиционно консервативных отраслях знаний, таких как: природа, биология, социология, экология, техника, транспорт и индустрия. Подтверждением сказанному могут служить модные глобальные технологии, использующие масштабируемые модели компьютеринга [2,3]: 1) Cyber-Physical Systems. 2) Internet of Things and Everything. 3) Web- Cloud-, Mobile-, Service-, Network-, Automotive, Big Data and Quantum computing. 4) Internet-Driven Smart Infrastructures: Enterprise, University, City and Government. Так, использование автоматной модели компьютеринга для описания процессов мозга привело к точному мониторингу нарушений в нем и последующему эффективному восстановлению функциональностей путем применения актюаторных воздействий [5]. Ученые предпринимают успешные попытки в разработке компьютерных структур на основе использования природных решений, отождествляемых с мозгом живых существ [6], что непременно приведет к созданию кибермозга человечества.

Понятие компьютеринга развивается из классической автоматной модели вычислителя, объединяющего механизмы управления и исполнения, сигналы мониторинга и актюации, входы для ввода инструкций и данных, а также выходы состояния системы и результатов.

Компьютинг – процесс достижения поставленной цели путем использования механизмов управления и исполнения в циклически замкнутой системе с заданными входами и выходами, сигналами мониторинга и актуации. В узком смысле компьютеринг – действия, целенаправленные на исследование, проектирование и применение интеллектуальных программно-аппаратных систем и сетей для мониторинга и управления киберфизическими процессами и явлениями. Область компьютеринга покрывает: нано-, микро- и макро-электронику, радиотелекоммуникации, компьютерную, программную, системную, производственную, транспортную и социальную инженерию, искусственный интеллект и киберуправление, компьютерные науки и информационные технологии.

Следует напомнить, что информационные технологии – процессы, методы и способы поиска, сбора, хранения, обработки и распространения информации путем использования средств вычислительной техники [7,8]. С учетом приведенных выше определений можно сделать вывод, что компьютеринг поглощает довольно устаревшее, но, в силу инерции человеческого мышления, распространенное понятие “информационные технологии”, которое формально отождествляется с отображением (мониторингом) физических процессов и явлений в виртуальное киберпространство.

Существенным представляется отличие компьютеринга от информационных технологий, связанное с активным управлением процессами и явлениями в реальном и виртуальном мирах. Можно поставить знак соответствия между понятиями содержания и формы: Internet = Information Technology и Internet of Things = Computing. Развитие компьютеринга имеет исторически выраженные фазы (см. рис. 1): 1) Сингулярный компьютеринг. 2) Сетевой компьютеринг. 3) Глобальный компьютеринг. 4) Киберфизический компьютеринг. 5) Сервис-компьютеринг – начало нового цикла. Далее предполагается появление киберчеловеческого компьютеринга, когда human brain будет непосредственно интегрирован с киберпространством. Развитие компьютеринга, основная функция которого – оптимальное и надежное управление всеми процессами и явлениями на основе точного цифрового мониторинга без прямого участия человека, следует рассматривать только во взаимодействии двух миров – реального и виртуального: 1) Человек всегда плохо управляет реальным миром и создает себе в помощь компьютеринг. 2) Как более совершенный механизм, компьютеринг забирает управление технологическими процессами у людей – фаза настоящего времени. 3) Чтобы спасти человечество от самоликвидации, компьютеринг, в ближайшее время, должен забрать и оставшуюся часть управления социальными процессами под свою юрисдикцию. 4) Человек и компьютеринг объединяются в желании творческого изменения киберфизического континуума в целях мирного сосуществования и взаимного проникновения для образования нового понятия киберчеловека (Cyber Human – CyMan). Киберчеловек – персона, владеющая глобальной технологической киберкультурой, непосредственно (Brain-Computing-Interface) подключенная к киберфизическому пространству планеты в целях исполнения киберсоциальной роли. Компьютеринг сегодня превращается из ненадежной локальной конструкции в неубиенную глобальную субстанцию. Человек постепенно будет трансформироваться из уязвимого биологического субъекта в кибербиологическую форму, а далее – в киберэнергетическую, «вечноживущую» информационно и физически реинкорнируемую субстанцию.

В результате взаимодействия человека с киберфизическим миром формируется новое понятие – киберкультура, как уровень развития социально-технологических отношений между обществом, физическим миром и киберпространством, определяемый внедрением интернет-сервисов точного цифрового мониторинга и надежного метрического управления во все процессы и сферы человеческой деятельности, включая образование, науку, медицину, производство и транспорт, в целях повышения качества жизни людей и сохранения экосистемы планеты.

Bernard Marr (Forbes) предложил 9 аттракторов [9], подтверждающих интерес рынка и мирового бизнеса к компьютерингу: Big Data. Internet of Things. Mobile to computing everywhere. Cyber security. E-Assistants or Brain-Computer Interface. Social Networks. Gamification for Business and Education. Cloud computing. Video communications. В ближайшем будущем ожидается массовое внедрение в жизнь людей новых технологий, формирующих Virical Continuum: Internet of Everything, wearable's, smart car, smart home, smart city, 3D printing,

quantum computing, robotics, the cloud, Big Data, the maker movement, drones. Особое внимание автор уделяет рекламе IoT, потому, что 87 процентов людей не слышали, что это такое. Фактически – это взаимодействие виртуального и реального мира: 2015 год – 5 миллиардов подключаемых устройств, 1,4 миллиарда смартфонов; 2020 – 50 миллиардов и 6,1 упомянутых изделий соответственно. Это также 250 миллионов машин в 2020 году, которые будут управляться без водителей с помощью облачных киберсервисов дорожного движения. Уже сегодня Google-машины проезжают порядка 10 000 миль в неделю по инновационным городским дорожным инфраструктурам. Более 10,2 миллиона «умной» одежды будет произведено к 2020 году. Также существенно возрастет рынок RFID-меток для цифровой идентификации объектов и процессов с 11,1 до 21,9 миллиарда долларов. В итоге, финансовое влияние IoT в 2025 году на мировой рынок составит сумму в 11 триллионов долларов, а уровень IoT-капитализации – 4,6 и 14,4 триллиона в публичном и приватном секторе соответственно.

Susan Galer (Forbes) считает, что облачный сервис будет наиболее интересной моделью IT-бизнеса в ближайшие 10 лет [10]. К концу 2017 года две трети от 2000 глобальных компаний трансформируют свою деятельность под цифровые процессы мониторинга и управления. Более 50 процентов всех инвестиций компьютерных (IT-) компаний будут направлены на создание технологических платформ и сервисов, связанных с cloud, mobile, social business and big-data analytics. К 2020 году cloud-based капитализация IT-компаний достигнет 70 процентов от всех программно-технологических сервисов. В 2018 году будут инсталлированы 22 миллиарда internet of things devices, которые будут иметь доступ к более, чем 200 000 новым internet of things приложениям и сервисам. К 2018 году более 50% компаний-разработчиков будут вставлять в свои приложения познавательные сервисы (против 1 процента сегодня), обеспечивая только USA-предприятиям более 60 миллиардов долларов ежегодной экономии. К этому времени 50 процентов всех предприятий создадут облачные платформы для распространения собственных инноваций и потребления внешних предложений. К 2018 году 80% компаний в формате B2B (Business-to-Business) и 60% B2C (Business-to-Customer) организаций перестроят свои входы (двери) на цифровые, что обеспечит увеличение потребителей и заказчиков на 3-4 порядка. Важным является и то, что к 2020 году более 30 процентов успешно существующих сегодня компаний – IT-поставщиков различных услуг, которые не перестроятся под новые цифровые отношения в киберпространстве, перестанут существовать. Можно резюмировать, что все предприятия и организации, включая университеты, города и страны, окажутся неконкурентоспособными без интеграции в киберпространство заранее подготовленными цифровыми правовыми отношениями (digital legislations).

## **2. МАТ-компьютинг для физического, виртуального и космологического пространства**

Цель – создание общей модели МАТ-компьютинга, как универсальной и масштабируемой структуры мониторинга и управления физическими, виртуальными и космологическими процессами и явлениями, предоставляющей человеку возможность осваивать новые жизненные пространства, включая космос, а также создавать рыночно-ориентированные продукты и киберсервисы, направленные на повышение качества жизни человека и сохранение экосистемы планеты [2,3,11-13].

Задачи: 1) Создание универсальной и масштабируемой модели МАТ-компьютинга, описывающей физические и виртуальные процессы мониторинга и управления. 2) Классификация существующих технологий компьютерного в реальном и виртуальном пространстве для создания общей картины киберфизического континуума (Cyber Physical Continuum). 3) Квантовая телепортация процессов, но не объектов, как технология проникновения человека в космос. 4) Киберфизический и квантовый компьютер, как методология описания процессов развития природы и общества.

Полезными могут быть модификации некоторых определений в сторону упрощения. Компьютинг – технология интерактивного мониторинга и управления процессами и явлениями для достижения цели по заданной программе без (прямого) участия человека. Основа компьютерного – транзакционное взаимодействие адресуемых данных (компонентов памяти) для достижения поставленной цели. Модель компьютерного: <Память, Адрес, Транзак-

ция> (рис. 3, слева). Память – субстанция, способная хранить данные (информацию). Адрес – субстанция, определяющая структуру координатами компонентов в виртуальном или реальном пространстве. Структура формируется различными субстанциями: гравитацией, внутри- и межатомным взаимодействием (базон Питера Хиггса), электромагнитными полями, гальваническими связями, сознанием человека, адресным взаимодействием. Транзакция – целенаправленный процесс приема-передачи данных между адресуемыми компонентами структурированной памяти для реализации функциональности или сервиса. Память – любая физическая субстанция, способная хранить, принимать и передавать данные, как виртуальную сущность. Поскольку любая материя состоит из атомов и электронов, то каждая из четырех форм существования любой материи обладает свойствами памяти.

Уравнение физического компьютеринга задает системное симметрическое (xor) взаимодействие трех равнозначных субстанций (Memory – Address – Transactions):  $MEAET=0$ . Это означает, что любой компонент определяется посредством взаимодействия двух других:  $M=AET$ ,  $A=TEM$ ,  $T=MEA$ . Виртуальный компьютеринг оперирует триадой: (Data – Address – Transactions). Универсальное характеристическое уравнение компьютеринга (компьютера) оперирует единственной адресной транзакцией (считывание-запись) данных между источником и приемником [2,3]:  $M(Y_i) = Q_i[M(X_i)]$ . Здесь  $W=<M,Q,X,Y>$  – структура, которая содержит адресуемые компоненты памяти и ничего более:  $M$  – память компьютеринга,  $Q$  – память функциональных примитивов – квантов [2,3],  $X$  – память адресов входных переменных примитивов,  $Y$  – память адресов выходных переменных примитивов. Естественно, что все компоненты памяти и ячейки в них являются адресуемыми. Это означает, что они могут заменяться и ремонтироваться в режиме remote and online. Недосток памяти один – проигрыш в скорости выполнения транзакций перед reusable logic или комбинационными логическими элементами. Таким образом, компьютеринг, как вычислительный процесс, сводится к пространственно-временной структуре транзакций над адресуемыми данными.

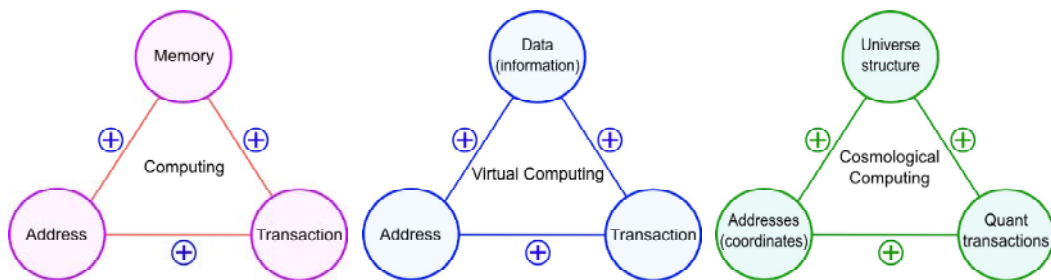


Рис. 3. Модель МАТ-компьютеринга

Далее представлены виды компьютеринга (рис. 4) без участия человека, но для человека и сохранения планеты, с указанием пары: тип носителя (память) – сигнатура (транзакции): 1) атомарный, состояние или спин электрона – квант, фотон; 2) кристаллический, состояние транзистора – электроны; 3) космологический, состояние материи – гравитационные и электромагнитные поля; 4) биологический, состояние клетки – электроны; 5) технологический, состояние системы – данные; 6) социологический, состояние общества – законодательство; 7) виртуальный (рис. 3, справа), состояние киберпространства – информация.

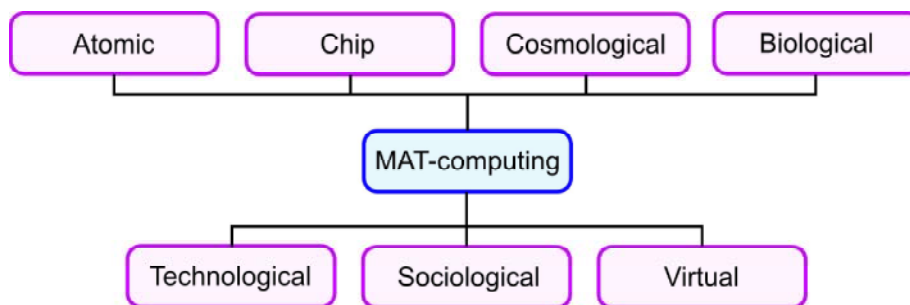


Рис. 4. Виды компьютеринга

Виртуальный или киберкомпьютинг оперирует данными (см. рис. 3, справа) вместо памяти, распределен в (кибер) пространстве и сегодня представлен модными структурными направлениями (рис. 5): 1) Cloud Computing. 2) Fog Network Computing. 3) Mobile Computing. 4) Service Computing. 5) Social Computing. 6) Automotive Computing. 7) Internet Computing – Smart Everything. 8) Cyber Physical- or Internet of Things (Everything) Computing. 9) Big Data Computing. 10) Quantum Computing (Security).

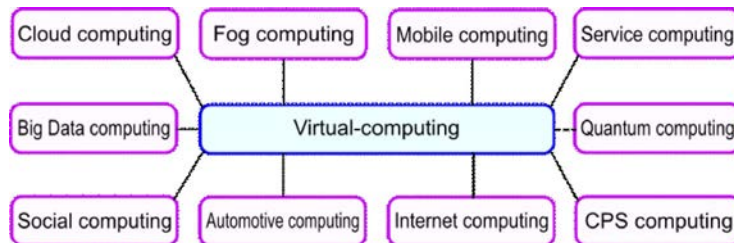


Рис. 5. Виртуальный (кибер) компьютеринг

Физический компьютеринг (рис. 6) сконцентрирован в функциональностях, сосредоточенных в компактном пространстве твердотельных специализированных устройств: 1) Quantum Computers. 2) Mobile Gadgets and Laptops. 3) Automotive Computers. 4) Smart Sensors and Actuators as MEMS. 5) Robotics. 6) Drones. 7) 3D-Printing. 8) Smart Brain-User Interfaces. 9) Security Computers. 10) Big Data Centers.

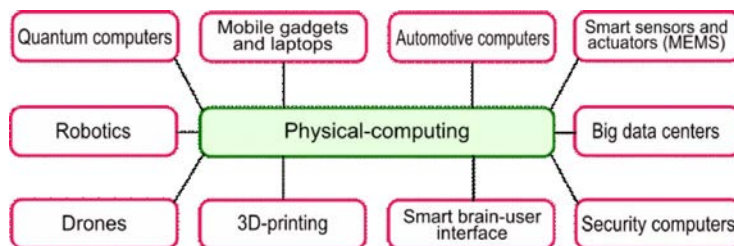


Рис. 6. Физический компьютеринг

Виды компьютеринга, представленные выше, покрываются обобщенной автоматной структурой управляемых физических и виртуальных процессов, известных человечеству (рис. 7).

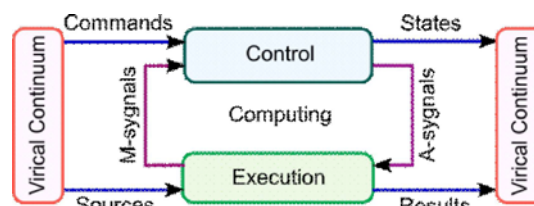


Рис. 7. Автоматная модель компьютеринга

Здесь в качестве основного компонента фигурирует память, как любая субстанция, способная хранить информацию, на которой организованы структурные компоненты компьютеринга (Control and Execution). Функционирование и развитие системы осуществляется посредством мониторинга (M-signals) и управления (актюации) (A-signals) компонентами вычислительной системы. Входы и выходы такой системы нагружены на единое внешнее виртуально-физическое (virical) пространство (virtual-physical continuum) [Forbes, Josh Linkner]. Такое континуальное определение киберфизического пространства относится не только к Земле, но и ко Вселенной, которая также имеет программу своего эволюционирования по самой простой и доступной для понимания автоматной модели компьютеринга. Модель

адекватно, идеально и просто описывает все процессы функционирования и эволюционирования технических, биологических, социальных, виртуальных и космологических объектов и структур на основе использования сигналов мониторинга и управления. При этом представленный выше автомат потребляет исходные данные и ресурсы из внешней экосистемы, которая также является местом, куда попадают результаты или продукты деятельности конкретного компьютеринга. Поэтому такой автомат, в зависимости от цели, может быть преобразователем кибер-физической экосистемы в Green Planet for Human или в сторону Armageddon деструктуризации.

### **3. Компьютеринг квантовой телепортации**

Компьютеринг квантовой телепортации подразумевает передачу на расстоянии не объектов, а технологических процессов и отношений, которые можно воссоздать на приемной стороне телекоммуникационного канала. Приход космических пришельцев на Землю не так уж невероятен. Сегодня созревает уверенность, что человечество сможет продолжить жизнь на подобных земным условиям планетах во Вселенной путем квантовой передачи актуаторных потоков информации, описывающих структуры и алгоритмы эволюционного синтеза и развития земных субстанций. Попав в благоприятную среду какой-нибудь планеты, кванто-кодированные геномы смогут подготовить инфраструктуру, а затем вырастить форсированным или эволюционным путем биологические и технические объекты. Осуществление такой дизрапторной экспансии человека в подходящую область космического пространства реально даже на современном уровне развития науки и технологий. При этом космические полеты с биологическими или техническими субстанциями на борту для межзвездных расстояний – тупиковый путь фантастов и ученых. Таким образом, расширение жизни человечества в космосе реально и произойдет именно путем квантовой телепортации процессов синтеза биотехнических объектов в приемлемую планетарную среду. Для этого ученые должны в земных условиях решить технологическую проблему квантовой актуации форсированного или эволюционного синтеза упомянутых субстанций. Нетрудно предположить, что более развитые цивилизации уже решили для себя проблему космической экспансии своих форм жизни и совершили квантовую интервенцию биотехнических объектов на Землю путем фотонной доставки алгоритмов их синтеза в благодатной земной среде.

Структурно-алгоритмическая сущность квантовой телепортации объекта заключается в доставке данных о связях компонентов в некоторую точку пространства, где имеются комплекующие элементы, а также о последовательности действий, составляющей оптимальную технологию изготовления некоторой структуры. Например, в Монголии можно построить фабрику по производству интегральных схем, для чего необходимо компании Интел нанять специалистов под телепортируемую в данную страну систему отношений, дать им технологии и технику для производства чипов. Чтобы создать университет мирового уровня (Кембридж, Стэнфорд, Массачусетс), нужно телепортировать в проблемный вуз только систему отношений (устав и традиции) от лидеров науки и образования. Чтобы создать благоприятную для населения страну, следует телепортировать гуманную человеко-ориентированную систему отношений (Конституцию), принятую, например, в США, Германии, Норвегии, Сингапуре.

Таким образом, при наличии абсолютно одинаковых условий и компонентов эффективность и качество структуры или системы определяются только связями или отношениями, равно как при наличии одинаковых кадров, зданий и целей эффективность университета определяется принятой в нем системой отношений (устав, положения, традиции). Все сказанное остается в силе и применимо к успешности городов и стран. Можно также добавить, что в случае неэффективности или ошибок в системе менять нужно не людей – глупое занятие – люди везде одинаковые, а структуру отношений между ними. *We are not looking for the people who made mistakes, but the situations where the mistakes have been possible (Stanley Hyduke).*

Квантовые (фотонные) транзакции космологического компьютеринга на структурированной гравитацией материи в форме звезд и планет, имеющих адреса в пространстве, расширяют горизонты жизни во Вселенной.

Квантовая структура биологического генома проникает в космос и создает новые формы жизни в других мирах. Человечество не обречено быть вечно на Земле. Гравитация с помощью бозона Хиггса создает структуры, а фотоны или кванты – транзакции на них. Структурные элементы Вселенной имеют адреса, выраженные через пространственно-временные координаты, благодаря гравитации. Бозон Хиггса ученые сегодня рассматривают как раствор для склеивания кирпичей мироздания в материальные формы и структуры. Гравитация – сила притяжения, вырабатываемая массами взаимодействующих космических тел – создает относительный структурный порядок во Вселенной на коротком временном интервале. Гравитации противодействует Большой взрыв, сила которого служит причиной всех изменений в цикле эволюции Вселенной до следующего Большого взрыва.

**Список литературы:** 1. *Бондаренко М.Ф., Хаханов В.И., Литвинова Е.И.* Структура логического ассоциативного мультипроцессора // Автоматика и телемеханика. Россия. 2012. С. 71-92. 2. *Хаханов В.И., Ваджеб Гариби, Литвинова Е.И., Шкиль А.С.* Кубитные структуры данных вычислительных устройств // Электронное моделирование. 2015. № 1. С. 76-99. 3. *Хаханов В.И., Тамер Бани Амер, Чумаченко С.В., Литвинова Е.И.* Кубитные технологии анализа и диагностирования цифровых устройств // Электронное моделирование. Том 37, № 3. С. 17-40. 4. *Vladimir Hahanov, Eugenia Litvinova, Wajeb Gharibi, Svetlana Chumachenko.* Big Data Driven Cyber Analytic System // 2015 IEEE Int. Congress on Big Data. New York. USA. P.615-622. 5. *Nathalie Dehorter, Gabriele Ciceri, Giorgia Bartolini, Lynette Lim, Isabel del Pino, and Oscar Marin.* Tuning of fast-spiking interneuron properties by an activity-dependent transcriptional switch. Science. 11 September 2015: 349 (6253). P. 1216-1220. 6. *Paul A. Merolla et al.* A million spiking-neuron integrated circuit with a scalable communication network and interface // Science, 8 August 2014: Vol. 345, no. 6197. P. 668-673. 7. *ГОСТ 34.003-90* Автоматизированные системы. Термины и определения. 8. ISO/IEC 38500:2008. Corporate governance of information technology. 9. *Bernard Marr.* Forbes. 9. *Attractors in the Computer Market.* 17 'Internet Of Things' Facts Everyone Should Read. <http://www.forbes.com/sites/bernardmarr/2015/10/27/17-mind-blowing-internet-of-things-facts-everyone-should-read/>. 10. *Susan Galer.* Forbes. IDC Releases Top Ten 2016 IT Market Predictions. <http://www.forbes.com/sites/sap/2015/11/05/idc-releases-top-ten-2016-it-market-predictions/> 11. *Бондаренко М.Ф., Хаханов В.И.* Квантовые технологии реализации мозгоподобных вычислительных структур академика В.М. Глушкова // Матеріали ХІХ Міжнародної конференції з автоматичного управління. Київ. 26-28 вересня, 2012. С. 32-33. 12. *Палагин А. В., Яковлев Ю. С., Тихонов Б. М., Першко И. М.* Архитектурно-структурная организация компьютерных средств класса "Процессор-в-памяти" // Математичні машини і системи. 2005. № 3. 14 с. 13. *Палагин А.В., Яковлев Ю.С.* Системная интеграция средств компьютерной техники. Винница: "Универсум", 2005. 14. <http://ru-universe.livejournal.com/712762.html?page=1>. 15. *Kelly K.F., Mody C.C.M.* The booms and busts of molecular electronics. 2015 //Spectrum, IEEE. Vol.52, Issue: 10. P. 52-60. 16. *Ллойд Сет.* Программируя Вселенную: Квантовый компьютер и будущее науки. М: Альпина нон-фикшн. 2014. 256 с.

*Поступила в редколлегию 23.12.2015*

**Tamer Bani Amer**, аспирант ХНУРЭ. Научные интересы: квантовые вычисления, тестирование и диагностика цифровых систем. Адрес: Украина, 61166, Харьков, пр. Науки, 14, тел. +3805770-21-326.

**Хаханов Владимир Иванович**, декан факультета КИУ ХНУРЭ, д-р техн. наук, проф. кафедры АПВТ ХНУРЭ, IEEE Senior Member, IEEE Computer Society Golden Core Member. Научные интересы: техническая диагностика цифровых систем, сетей и программных продуктов. Увлечения: баскетбол, футбол, горные лыжи. Адрес: Украина, 61166, Харьков, пр. Науки, 14, тел. +3805770-21-326. E-mail: [hahanov@kture.kharkov.ua](mailto:hahanov@kture.kharkov.ua).

**Емельянов Игорь Валерьевич**, н.с. кафедры АПВТ ХНУРЭ. Адрес: Украина, 61166, Харьков, пр. Науки, 14, тел. +3805770-21-326.

**Любарский Михаил**, аспирант кафедры АПВТ ХНУРЭ. Адрес: Украина, 61166, Харьков, пр. Науки, 14, тел. +3805770-21-326.

## ЭФФЕКТИВНОЕ КОДИРОВАНИЕ СЛОТОВ Р-КАДРОВ В ПРОСТРАНСТВЕ СТРУКТУРНО-ЛОКАЛЬНЫХ ОГРАНИЧЕНИЙ ДЛЯ ПОВЫШЕНИЯ БЕЗОПАСНОСТИ ДИНАМИЧЕСКИХ ВИДЕОРЕСУРСОВ

---

Обосновывается необходимость повышения безопасности динамических видеоинформационных ресурсов в системах управления и объективного контроля для стратегически значимых отраслей и ведомственных организаций. Излагается подход для создания метода обработки последовательности Р-кадров на основе сокращения межтрансформантной избыточности без потери целостности информации. Формулируются требования, которые необходимо заложить в процесс разработки эффективного синтаксического описания информативной дифференциально описанной спектрограммы как базовой структурной единицы слотов Р-кадров. Излагается правило индексирования информативных дифференциально описанных спектрограмм с учетом количества допустимых последовательностей в сформированном градиентном пространстве. Разрабатывается эффективное кодирование нормированной информативной дифференциально описанной спектрограммы в двухкомпонентном градиентном пространстве структурно-локальных ограничений для повышения безопасности динамических видеоресурсов.

### 1. Введение

Эффективность функционирования стратегически значимых для государства отраслей и ведомственных организаций во многом зависит от уровня информатизации и обеспечения безопасности информации. Информационная безопасность определяется тремя основными категориями: доступность, целостность и конфиденциальность. В кризисных ситуациях наибольший вес имеют угрозы потери целостности и доступности информационного ресурса [1, 2]. Существуют угрозы нарушения свойств доступности и целостности. Такая ситуация происходит вследствие наличия множества уязвимостей, обусловленных ограниченными характеристиками инфокоммуникационных технологий, в том числе использующих тракт беспроводной передачи данных. Это особенно критично в случае необходимости реализации видеоинформационного взаимодействия или организации сбора видеоинформации с дистанционных сенсоров.

При такой ситуации неминуемы задержки по доставке информации. Для решения сложившейся проблемы интегрируются технологии эффективного синтаксического представления видеопотоков [3, 4]. В то же время существующие технологии обработки имеют недостатки. Снижение интенсивности битового потока достигается ценой увеличения задержек на время обработки и потери целостности информации. Поэтому исследования, касающиеся совершенствования технологий эффективного синтаксического представления видеоданных, являются актуальными.

Структурный анализ последовательности кадров изображений показал наличие в них высокой избыточности. Она обусловлена присутствием в соседних кадрах областей стационарного фона. Информацию о стационарных областях допускается передавать с использованием механизмов структурной обработки. Следовательно, в условиях формирования изображений стационарного фона резко повышается количество временной избыточности, обусловленной наличием межкадровых структурных закономерностей. Учет такой особенности не предусмотрен для существующих кодеков видеопотока. Поэтому предлагается применять подход, базирующийся на обработке последовательности кадров с использованием структурного подхода, учитывающего наличие для последовательности кадров областей стационарного фона [4]. Отсюда *цель исследований* заключается в разработке эффективного кодирования слотов Р-кадров в градиентном пространстве структурно-локальных ограничений для повышения безопасности динамических видеоресурсов.

## 2. Формирование концептуального базиса эффективного синтаксического представления слотов R-кадров

Основной структурной единицей (структурным базисом) для разработанного подхода обработки последовательности кадров в потоке является дифференциально описанная спектрограмма (ДОС)  $E^{(k,\ell)}$ . В случае привязки ДОС к слоту формируется запись  $E(\chi;\gamma)^{(k,\ell)}$  –  $(k;\ell)$ -я дифференциально описанная спектрограмма для  $(\chi;\gamma)$ -го слота последовательности ДОТ кадров. Поскольку слот  $E(T-1)_{\chi,\gamma}^{(2)}$  трактуется как двумерный пакет дифференциально описанных спектрограмм  $E(\chi;\gamma)^{(k,\ell)}$ , то его можно представить следующей записью [5]:

$$E(T-1)_{\chi,\gamma}^{(2)} = \bigcup_{k=1}^w \bigcup_{\ell=1}^w E(\chi;\gamma)^{(k,\ell)}. \quad (1)$$

С другой стороны, по определению слот  $E(T-1)_{\chi,\gamma}^{(2)}$  представляет собой сквозную квадратную трубу, вырезанную вдоль последовательности ДОТ кадров на позиции  $(\chi;\gamma)$ .

Тогда с учетом соотношения (1) группу  $Y_T$  трансформированных кадров  $Y(\tau)$ ,  $Y_T = \{Y(\tau)\}$ ,  $\tau = \overline{1, T}$  можно выразить через ДОС:

$$Y_T = Y(1) \cup E_{T-1} = Y(1) \cup \bigcup_{\chi=1}^{w_{\text{стр}}} \bigcup_{\gamma=1}^{w_{\text{стб}}} \bigcup_{k=1}^w \bigcup_{\ell=1}^w E(\chi;\gamma)^{(k,\ell)}.$$

Здесь  $Y(1)$  – базовый трансформированный кадр.

В свою очередь каждая ДОС  $E(\chi;\gamma)^{(k,\ell)}$  представляется в виде совокупности информативной  $E(\chi;\gamma)_{\text{inf}}^{(k,\ell)}$  и интерполированной  $E(\chi;\gamma)_{\text{int}}^{(k,\ell)}$  частей, т.е.

$$E(\chi;\gamma)^{(k,\ell)} = E(\chi;\gamma)_{\text{inf}}^{(k,\ell)} \cup E(\chi;\gamma)_{\text{int}}^{(k,\ell)}.$$

При этом базисом для эффективного синтаксического описания информативной ДОС является двухкомпонентное градиентное ограничение, а именно  $\{\bar{d}_e^{(k,\ell)}; g(\max)^{(k,\ell)}\}$ .

Величина обобщенного градиентного ограничения  $g(\max)^{(k,\ell)}$  вычисляется для каждой информативной дифференциально описанной спектрограммы по формуле

$$g(\max)^{(k,\ell)} = \max_{1 \leq \tau \leq n_{\text{inf}}} \{ |e(\tau+1)_{\text{inf}}^{(k,\ell)} - e(\tau)_{\text{inf}}^{(k,\ell)}| \},$$

где  $e(\tau)_{\text{inf}}^{(k,\ell)}$ ,  $e(\tau+1)_{\text{inf}}^{(k,\ell)}$  – информативные элементы  $(k;\ell)$ -й интерполированной ДОС, соответственно расположенные в ней на  $\tau$ -й и  $(\tau+1)$ -й позициях.

С учетом этого, для первого информативного элемента ДОС динамический диапазон  $\bar{d}(1)_{\text{inf}}^{(k,\ell)}$  будет равен общему динамическому диапазону  $\bar{d}_e^{(k,\ell)}$  последовательности  $E_{\text{inf}}^{(k,\ell)}$ .

Величина  $\bar{d}(\tau)_{\text{inf}}^{(k,\ell)}$  динамического диапазона остальных элементов информативной дифференциально описанной спектрограммы,  $\tau = \overline{2, n_{\text{inf}}}$ , будет равна

$\bar{d}(1)_{\text{inf}}^{(k,\ell)} = \bar{d}_e^{(k,\ell)} = |e_{\text{min}}^{(k,\ell)} - e_{\text{max}}^{(k,\ell)}| + 1$ . Отсюда на максимальные абсолютные значения (спецификации) элементов  $e(\tau)_{\text{inf}}^{(k,\ell)}$  последовательности  $E(\chi;\gamma)_{\text{inf}}^{(k,\ell)}$  в градиентном пространстве  $G^{(k,\ell)}$  наложены ограничения, описываемые следующей системой неравенств:

$$|e(\tau)_{\text{inf}}^{(k,\ell)}| = \begin{cases} \bar{d}_e^{(k,\ell)}, & \rightarrow \tau=1; \\ 2g(\text{max})^{(k,\ell)} + 1, & \rightarrow 2 \leq \tau \leq n_{\text{inf}}. \end{cases} \quad (2)$$

Поэтому для повышения безопасности динамических ВИР необходимо синтезировать функционал  $F(E^{(k,\ell)})_{\text{apr}}$ , задающий эффективное синтаксическое представление информативной ДОС  $E(\chi; \gamma)_{\text{inf}}^{(k,\ell)}$  как базовой структурной единицы слотов Р-кадров в градиентном пространстве локально-структурных закономерностей. При этом необходимо учитывать следующие особенности:

1) выявление структурно-локальных закономерностей проводится по результату анализа всех элементов ДОС;

2) ограниченные значения динамических диапазонов элементов ДОС в соответствии с закономерностями, описанными системой неравенств (2);

3) длина  $n_{\text{inf}}$  информативной дифференциально-описанной спектрограммы устанавливается заранее до начала процесса обработки;

4) необходимость восстановления элементов ДОС в условиях обеспечения требуемого уровня целостности динамических ВИР. Здесь требуется учитывать то, что информативные элементы  $e(\tau)_{\text{inf}}^{(k,\ell)}$  ДОС несут информацию об элементах  $e(\tau)_{\text{int}}^{(k,\ell)}$  ее интерполированной составляющей  $E(\chi; \gamma)_{\text{int}}^{(k,\ell)}$ , а величина ошибки зависит от значения обобщенной характеристики  $g(\text{max})^{(k,\ell)}$  градиентного пространства. Следовательно, информативные элементы должны обрабатываться без потери целостности;

5) количество  $Q(n_{\text{inf}}; g(\text{max})^{(k,\ell)})_e$  комбинаций для информативной ДОС в условиях, когда ее элементы удовлетворяют системе закономерностей (2), в двухкомпонентном градиентном базисе  $\{\bar{d}_e^{(k,\ell)}; g(\text{max})^{(k,\ell)}\}$  определяется по формуле:

$$Q(n_{\text{inf}}; g(\text{max})^{(k,\ell)})_e = \bar{d}_e^{(k,\ell)} \cdot (2g(\text{max})^{(k,\ell)} + 1)^{n_{\text{inf}} - 1}. \quad (3)$$

Это позволяет рассматривать информативную ДОС  $E(\chi; \gamma)_{\text{inf}}^{(k,\ell)}$  как позиционное число с весовыми коэффициентами, вычисляемыми с использованием соотношения (3);

6) элементы  $e(\tau)_{\text{inf}}^{(k,\ell)}$  информативной ДОС принимают как положительные, так и отрицательные значения.

Исходя из этого, для формирования эффективного синтаксического представления без формирования дополнительной (избыточной) служебной информации о знаках элементов ДОС предлагается провести их нормировку в область положительных значений. Такой процесс в дальнейшем будем называть градиентной нормализацией.

Согласно изложенному можно заключить, что сформулированы требования, которые необходимо заложить в процесс разработки эффективного синтаксического описания информативной ДОС как базовой структурной единицы слотов Р-кадров для повышения безопасности ДВИР.

### **3. Разработка эффективного синтаксического представления дифференциально описанной спектрограммы в составе слотов Р-кадров**

В соответствии с пятой особенностью информативной ДОС формировать эффективное синтаксическое представление предлагается с учетом ее интерпретации как позиционного числа. В связи с этим требуется установить правило генерирования индексов (индексирование). Индексирование последовательностей  $E(\chi; \gamma)_{\text{inf}}^{(k,\ell)}$  предлагается задавать следующим образом:

$$Q(E(\chi; \gamma)_{\text{inf}}^{(k, \ell)})_e = C(E(\chi; \gamma)_{\text{inf}}^{(k, \ell)})_e = \sum_{\tau=1}^{n_{\text{inf}}} \sum_{\eta=e_{\text{min}}^{(k, \ell)}}^{e(\tau)_{\text{inf}}^{(k, \ell)}-1} Q(e(1)_{\text{inf}}^{(k, \ell)}, e(2)_{\text{inf}}^{(k, \ell)}, \dots, e(\tau-1)_{\text{inf}}^{(k, \ell)}, \eta), \quad (4)$$

где  $C(E(\chi; \gamma)_{\text{inf}}^{(k, \ell)})_e$  – индекс последовательности  $E(\chi; \gamma)_{\text{inf}}^{(k, \ell)}$  во множестве допустимых информативных ДОС с учетом выявленного градиентного пространства;

$Q(e(1)_{\text{inf}}^{(k, \ell)}, e(2)_{\text{inf}}^{(k, \ell)}, \dots, e(\tau-1)_{\text{inf}}^{(k, \ell)}, \eta)$  – количество информативных дифференциально описанных спектрограмм, у которых первые  $\tau$  информативных элементов равны соответственно  $(e(1)_{\text{inf}}^{(k, \ell)}, e(2)_{\text{inf}}^{(k, \ell)}, \dots, e(\tau-1)_{\text{inf}}^{(k, \ell)}, \eta)$ .

Соотношение (4) позволяет провести индексацию информативных ДОС с учетом количества допустимых последовательностей в сформированном градиентном пространстве, которое образуется для разного количества информативных элементов.

*Теперь рассмотрим процесс градиентной нормализации.* Данный процесс заключается в сдвиге отрицательной области значений информативного элемента  $e(\tau)_{\text{inf}}^{(k, \ell)}$  в положительном направлении симметрично относительно нулевого уровня.

Тогда образуются величины  $\bar{e}(\tau)_{\text{inf}}^{(k, \ell)}$ , равные

$$\bar{e}(\tau)_{\text{inf}}^{(k, \ell)} = (e(\tau+1)_{\text{inf}}^{(k, \ell)} - e(\tau)_{\text{inf}}^{(k, \ell)} + g(\text{max})^{(k, \ell)}).$$

Отсюда замечаем, что величины  $\bar{e}(\tau)_{\text{inf}}^{(k, \ell)}$  принимают положительные значения в диапазоне  $0 \leq \bar{e}(\tau)_{\text{inf}}^{(k, \ell)} \leq 2g(\text{max})^{(k, \ell)}$ . Соответственно, информативная ДОС  $E(\chi; \gamma)_{\text{inf}}^{(k, \ell)}$  после градиентной нормализации представляется нормированной последовательностью  $\bar{E}(\chi; \gamma)_{\text{inf}}^{(k, \ell)}$ , т.е.  $\bar{E}(\chi; \gamma)_{\text{inf}}^{(k, \ell)} = \{\bar{e}(\tau)_{\text{inf}}^{(k, \ell)}\}$ ,  $\tau = \overline{1, n_{\text{inf}}}$ .

Рассмотрим процесс разработки эффективного синтаксического представления нормированной информативной ДОС  $\bar{E}(\chi; \gamma)_{\text{inf}}^{(k, \ell)}$  с учетом сформулированного правила индексирования.

Для этого заметим, что количество  $Q(\bar{e}(1)_{\text{inf}}^{(k, \ell)}, \bar{e}(2)_{\text{inf}}^{(k, \ell)}, \dots, \bar{e}(\tau-1)_{\text{inf}}^{(k, \ell)})$  нормированных информативных ДОС, у которых первые  $\tau$  информативных элементов равны соответственно  $(\bar{e}(1)_{\text{inf}}^{(k, \ell)}, \bar{e}(2)_{\text{inf}}^{(k, \ell)}, \dots, \bar{e}(\tau-1)_{\text{inf}}^{(k, \ell)})$ , а оставшиеся  $(n_{\text{inf}} - \tau)$  информативных элементов принимают значения согласно системе (2), находится по определению в соответствии с выражением (3). Получим следующее выражение:

$$Q(\bar{e}(1)_{\text{inf}}^{(k, \ell)}, \bar{e}(2)_{\text{inf}}^{(k, \ell)}, \dots, \bar{e}(\tau-1)_{\text{inf}}^{(k, \ell)}) = (2g(\text{max})^{(k, \ell)} + 1)^{n_{\text{inf}} - \tau}.$$

После этого проведя суммирование по всем  $\tau$ , где  $\tau = \overline{1, n_{\text{inf}}}$ , получим

$$\begin{aligned} C(\bar{E}(\chi; \gamma)_{\text{inf}}^{(k, \ell)})_e &= \sum_{\tau=1}^{n_{\text{inf}}} \bar{e}(\tau)_{\text{inf}}^{(k, \ell)} Q(\bar{e}(1)_{\text{inf}}^{(k, \ell)}, \bar{e}(2)_{\text{inf}}^{(k, \ell)}, \dots, \bar{e}(\tau-1)_{\text{inf}}^{(k, \ell)}) = \\ &= \sum_{\tau=1}^{n_{\text{inf}}} \bar{e}(\tau)_{\text{inf}}^{(k, \ell)} (2g(\text{max})^{(k, \ell)} + 1)^{n_{\text{inf}} - \tau}. \end{aligned}$$

Данное соотношение обеспечивает формирование кодового значения  $C(\bar{E}(\chi; \gamma)_{\text{inf}}^{(k, \ell)})_e$  эффективного синтаксического представления для нормированной информативной ДОС  $\bar{E}(\chi; \gamma)_{\text{inf}}^{(k, \ell)}$ , элементы которой удовлетворяют системе ограничений (2), а их количество равно  $n_{\text{inf}}$ .

По изложенному материалу можно сделать такие заключения:

1) создано правило индексирования информативных ДОС с учетом количества допустимых последовательностей в сформированном градиентном пространстве, которое образуется для разного количества информативных элементов;

2) разработано эффективное кодирование нормированной информативной дифференциально описанной спектрограммы в двухкомпонентном градиентном пространстве структурно-локальных ограничений для повышения безопасности динамических ВИР.

#### 4. Выводы

Разработано эффективное кодирование нормированной информативной дифференциально описанной спектрограммы в двухкомпонентном градиентном пространстве структурно-локальных ограничений для повышения безопасности динамических ВИР, учитывающее: выявление структурно-локальных закономерностей по результату анализа всех элементов ДОС; ограниченные значения динамических диапазонов элементов ДОС в соответствии с закономерностями двухкомпонентного градиентного пространства; то, что длина информативной дифференциально описанной спектрограммы устанавливается заранее до начала процесса обработки; необходимость восстановления информативных элементов ДОС без потери целостности; рассмотрение информативной ДОС как позиционного числа с весовыми коэффициентами, вычисляемыми с использованием ограничений градиентного пространства; исключение дополнительной (избыточной) служебной информации о знаках элементов ДОС путем процесса градиентной нормализации.

*Научная новизна результатов исследований.* Впервые синтезирован метод эффективного синтаксического представления последовательностей Р-кадров на основе межтрансформантного кодирования. Отличия от существующих методов заключаются в том, что кодовое значение формируется для нормированной информативной дифференциально описанной спектрограммы в составе слотов Р-кадров с учетом двухкомпонентных градиентных ограничений локально-структурных закономерностей. Это позволяет снизить битовую интенсивность закрытой по базовому кадру группы кадров без потери целостности информации.

**Список литературы:** 1. *Комарова Л.О.* Методи управління інформаційно-комунікаційними кластерами в кризових ситуаціях: монографія [Текст] / Л.О.Комарова // К.:ДУТ, 2014. 395 с. 2. *Автоматизированная система коммерческого осмотра поездов и вагонов* / Под ред. В.Н. Солошенко. М.: ГОУ „Учебно-методический центр по образованию на железнодорожном транспорте”. 2008. 3. *Ватолин Д., Ратушняк А., Смирнов М., Юкин В.* Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео М.: Диалог-Мифи, 2003. 381с. 4. *Баранник В.В.* Кодирование трансформированных изображений в инфокоммуникационных системах / В.В. Баранник, В.П. Поляков. Х.: ХУПС, 2010. 212 с. 5. *Баранник В.В.* Модель оценки информативности слота Р-кадров на основе выявления структурно-градиентных межтрансформантных ограничений / В.В. Баранник, С.С. Шульгин // АСУ и приборы автоматики. 2015. Вып. 172. С. 76-81.

*Поступила в редколлегию 28.12.2015*

**Баранник Владимир Викторович**, д-р техн. наук, профессор, начальник кафедры Харьковского университета Воздушных Сил имени Ивана Кожедуба. Научные интересы: системы, технологии преобразования, кодирования, защиты и передачи информации, семантической обработки изображений. Адрес: Украина, 61023, Харьков, 23, ул. Сумская, 77/79, тел. 8-050-3038971.

**Шульгин Сергей Сергеевич**, соискатель Киевского Национального авиационного университета. Научные интересы: обработка видеoinформации и безопасность инфокоммуникационных систем. Адрес: Украина, Киев, пр. Космонавта Комарова, 1.

## **ОБОСНОВАНИЕ ПОДХОДА ДЛЯ ФОРМИРОВАНИЯ КВАНТОВАННОГО ОПИСАНИЯ ТРАНСФОРМАНТЫ СЕГМЕНТА АЭРОФОТОСНИМКА**

---

Рассматриваются существующие технологии обработки изображений с применением структурного кодирования. Указывается на наличие противоречия между необходимостью уменьшать информационную интенсивность для фрагментов с сильной насыщенностью и поддержания соответствия восстановленных фрагментов исходному аэрофотоснимку. Предлагается производить квантование описания трансформанты сегмента аэрофотоснимка по отношению значения элемента вектора оснований к некоторому пороговому значению. Обосновывается сокращение комбинаторной избыточности в результате применения представления массива верхнего квантованного уровня в раздельном пространстве.

### **1. Введение**

В целях улучшения качества системы управления в государственных структурах происходит активное внедрение систем наблюдения [1], поэтому в ведомственных организациях, особенно в Министерстве обороны Украины, широко распространяются системы аэромониторинга с использованием беспилотных летательных аппаратов. В результате наблюдения получают видовые изображения местности с помощью аэрофотографирования. В силу важности содержания полученного изображения возникает необходимость максимально снизить временные затраты на обработку и доведение данных аэрофотоснимка в условиях поддержки необходимого уровня соответствия восстановленного изображения исходному [2]. В процессе обработки аэрофотоснимков в разнообразных специализированных устройствах применяют математическое описание изображения как двумерного сигнала, который характеризуется объемом информации [3].

Чтобы обосновать вариант решения поставленной задачи, необходимо рассмотреть существующие технологии обработки аэрофотоснимков. Такие технологии как JPEG, JPEG2000 используют сжатие на базе статистического кодирования (кодов переменной длины) [4]. Для данных технологий не представляется возможным определить до момента приема потока данных значение информационной интенсивности. Напротив, в случае применения структурного кодирования из служебной информации о векторе оснований путем формирования вектора-столбца взвешенных коэффициентов задается максимальное значение и соответственно длина кода-номера. Такая схема кодирования приводит к снижению количества разрядов на кодовое представление массивов видеоданных. Важным свойством полиадических чисел является возможность вычисления кода для массива данных на основе аналитического выражения. При этом необходимо знать только значения элементов массива данных и значения компонент вектора оснований. Поэтому не требуется организовывать формирование таблицы допустимых чисел и проводить поиск в таблице необходимого индекса, соответствующего обрабатываемому числу.

В то же время возникает проблема уменьшения временных затрат на доставку данных аэрофотоснимка с сильнонасыщенными фрагментами, которые характеризуются повышенной информационной интенсивностью.

Цель работы заключается в обосновании подхода к формированию квантованного описания трансформанты сегмента аэрофотоснимка.

### **2. Основной материал**

В результате обнаружения зависимостей неравномерностей динамических диапазонов и их ограниченности для аэрофотоснимков с различными параметрами структурной сложности предоставляется возможность уменьшить количество статистической избыточности

путем применения структурно-блочного кодирования в методе обработки аэрофотоснимков [5].

Для этого определяются параметры обработки при выполнении нормирования на основе параметров структурной сложности. Это приводит к образованию спектрального раздельно-нормированного пространства (СРНП) [6].

Значения компонент трансформанты  $X^{(W)}$  характеризуются динамическим диапазоном компонент  $x_{k\ell}$  в двухмерном пространстве с учетом диапазонов строк  $\Delta_k$  и столбцов  $\Delta_\ell$  как величины  $\Delta_{k\ell}$  :

$$\Delta_{k\ell} = \min ( x_{\ell, \max} + 1; x_{k, \max} + 1) ,$$

где  $\Delta_{k\ell}$  – величина диапазона компоненты;  $x_{k\ell}, \Delta_{\ell, \max}$  – величина диапазона  $\ell$ -го столбца трансформанты  $X$ ;  $\Delta_{k, \max}$  – величина диапазона  $k$ -й строки трансформанты  $X$ .

Для дополнительного сокращения объемов сжатых аэрофотоснимков предлагается использовать их квантование, которое образуется при выявлении компонент трансформанты с высоким динамическим диапазоном значений (рис. 1).

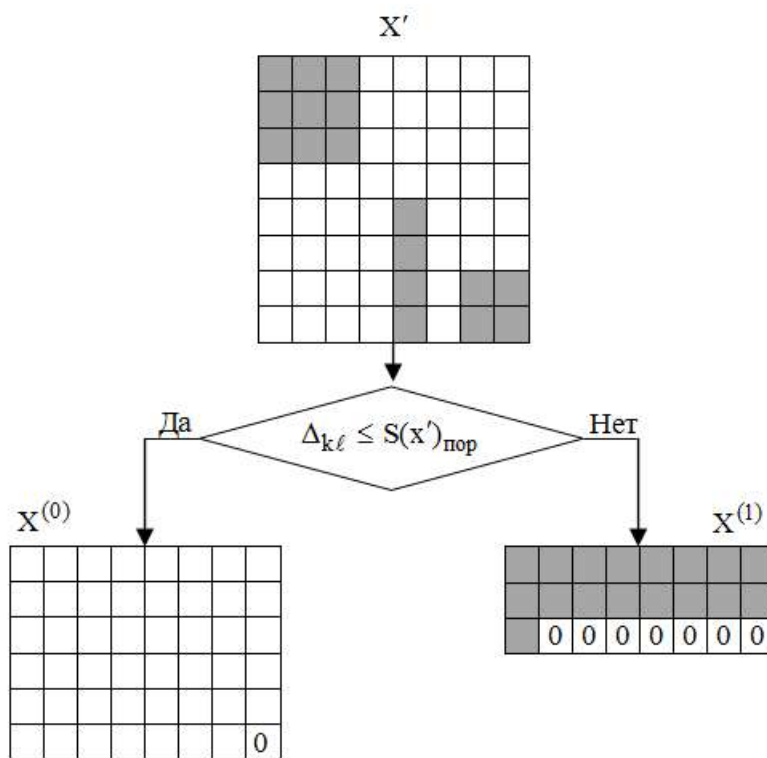


Рис. 1. Структурно-функциональная схема квантования массива раздельного представления

При квантовании происходит распределение исходного массива раздельного представления  $X' = \{x'_{k\ell}\}$  на две составляющие  $X^{(0)}$  и  $X^{(1)}$ . Принадлежность к данным составляющим определяется отношением величины  $\Delta_{k\ell}$  одному из двух уровней динамического диапазона массива дифференциального представления, т.е.

$$X \xrightarrow{S(x')_{\text{пор}}} \{X^{(0)}; X^{(1)}\},$$

где  $X^{(0)}, X^{(1)}$  – составляющие массива раздельного представления, в состав которых входят соответственно элементы нижнего и верхнего квантованных уровней.

Квантование в массивах дифференциального представления возникает вследствие комбинированного подхода в процессе его формирования. В результате этого формируется динамический диапазон массива раздельного дифференциального представления, структура которого имеет минимум две градации. Из этого следует, что при квантовании происходит устранение структурной избыточности.

Для этого предлагается проводить квантование массивов раздельного представления (МРП).

В этом случае возможно устранение недостатка, причина возникновения которого состоит в высоком динамическом диапазоне отдельных элементов массива дифференциального представления. Это достигается благодаря следующим характеристикам составляющей  $X^{(1)}$  верхнего квантованного уровня:

- в массив верхнего диапазонного уровня квантованного представления входят элементы, которым соответствуют перепады на границе контура и основного фона фрагмента аэрофотоснимка;
- для элементов, которым соответствует граница контура и основного фона фрагмента аэрофотоснимка, характерный однородный динамический диапазон.

Данные характеристики используются в раздельном описании составляющей верхнего квантованного уровня. При раздельном представлении массивов верхнего диапазонного уровня элементы массива представляются как двумерные позиционные числа в разностном пространстве. При этом код массива  $X^{(1)}$  будет определяться относительно кода числа, которое соответствует минимальному уровню  $V^{(2)}$  раздельного пространства (рис. 2).

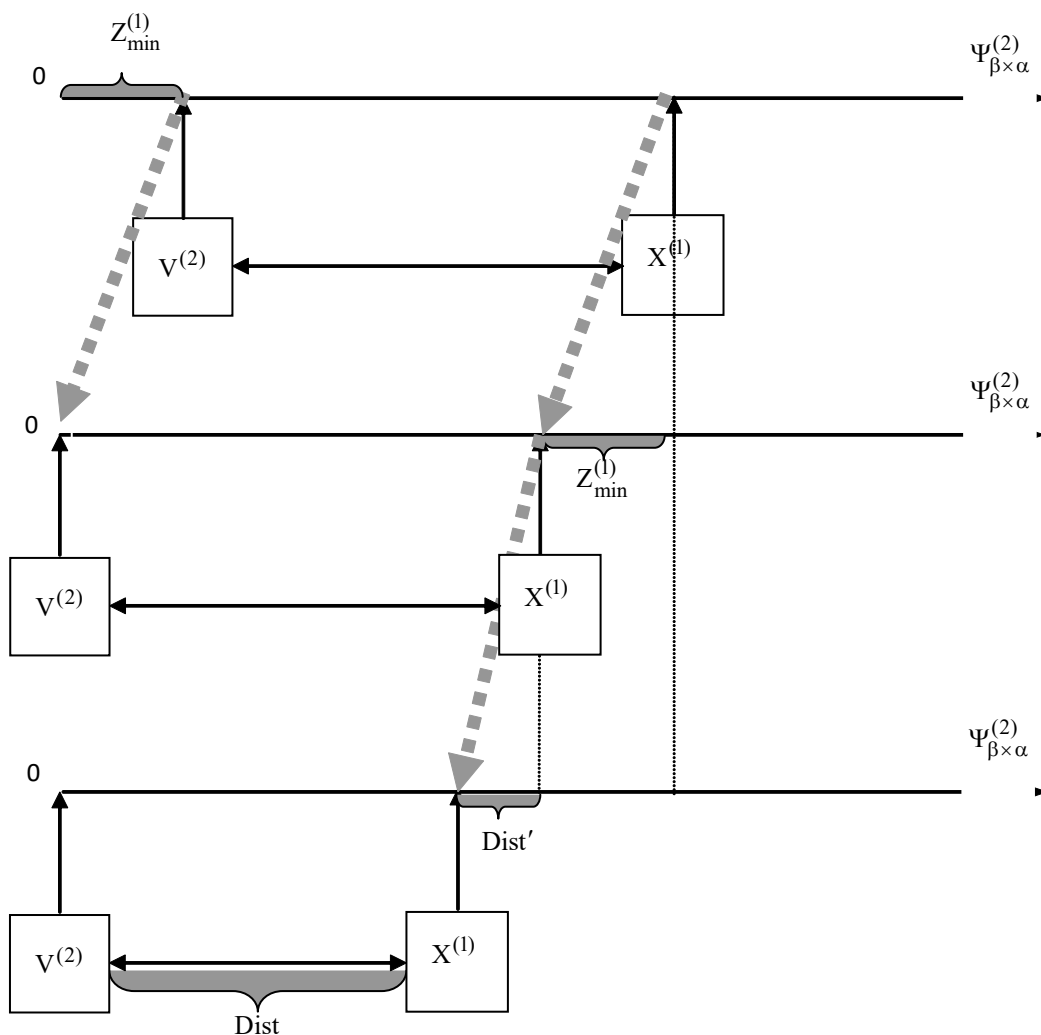


Рис. 2. Схема формирования элементов двумерного дифференциального неравновесного позиционного числа

В связи с этим предлагается учитывать минимальные значения в каждой строке массива верхнего квантованного уровня в процессе формирования минимального уровня раздельного пространства. При этом образуется вектор минимальных значений в строках,  $V = \{v_1, \dots, v_\beta\}$ . Величина  $v_i$  рассчитывается как минимальное значение в  $i$ -й строке массива  $X^{(1)}$ , т.е.  $v_i = \min_{1 \leq j \leq \beta} \{X_{ij}^{(1)}\}$ ,  $i = \overline{1, \alpha}$ .

Определение минимальных значений для каждой строки позволяет учесть особенности динамических диапазонов на границах объектов аэрофотоснимка.

В результате этого достигается смещение от исходного динамического диапазона массива  $X^{(1)}$  к пониженному динамическому диапазону, который задается вектором ограничений  $Dist$ , т.е.  $Dist = \{dist_{1j}, \dots, dist_{\beta j}\}$ . Здесь  $dist_{ij}$  – разность между максимальным  $\Delta_{ij}^{(1)}$  и минимальным  $v_i$  значениями в  $i$ -й строке массива  $X^{(1)}$  верхнего квантованного уровня, т.е.  $dist_{ij} = \Delta_{ij}^{(1)} - v_i$ .

Нижний уровень  $V^{(2)}$  двумерного раздельного пространства рассчитывается, базируясь на значениях элементов вектора  $V = \{v_1, \dots, v_m\}$ . Элементы нижнего уровня  $V^{(2)}$  задаются ограничениями на динамический диапазон верхнего квантованного уровня.

Принимая во внимание выражение (2), значение расстояния  $Dist$  между текущим массивом  $X^{(1)}$  и нижним уровнем  $V^{(2)}$  двумерного числа рассчитывается согласно следующей формуле:

$$Dist = \sum_{i=1}^{\alpha} \sum_{j=1}^{\beta^{(1)}} (x_{ij}^{(1)} - v_i) \prod_{\xi=j+1}^{\beta^{(1)}} dist_{i\xi} \alpha \prod_{\gamma=i+1}^{\beta} \prod_{\xi=1}^{\alpha^{(1)}} dist_{\gamma\xi},$$

где  $(x_{ij}^{(1)} - v_i) = \bar{x}_{ij}^{(1)}$  – значение смещения элементов текущего массива в нижний уровень двумерного ДНП, индекс позиции которых равняется  $(i; j)$ .

Обоснуем необходимость представления массива  $X^{(1)}$  в двумерном раздельном пространстве. Для этого распишем значение кода  $Z^{(1)}$  в исходном пространстве:

$$Z^{(1)} = \sum_{i=1}^{\alpha} \sum_{j=1}^{\beta^{(1)}} x_{ij}^{(1)} \prod_{\xi=j+1}^{\beta^{(1)}} \Delta_{i\xi}^{(1)} \prod_{\gamma=i+1}^{\alpha} \prod_{\xi=1}^{\beta^{(1)}} \Delta_{\gamma\xi}^{(1)}.$$

$$Dist = \sum_{i=1}^{\alpha} \sum_{j=1}^{\beta^{(1)}} (x_{ij}^{(1)} - v_i) \prod_{\xi=j+1}^{\beta^{(1)}} (\Delta_{i\xi}^{(1)} - v_i) \prod_{\gamma=i+1}^{\alpha} \prod_{\xi=1}^{\beta^{(1)}} (\Delta_{\gamma\xi}^{(1)} - v_\gamma).$$

Поскольку выполняются неравенства:

$$(x_{ij}^{(1)} - v_i) \leq x_{ij}^{(1)};$$

$$\prod_{\xi=j+1}^{\beta^{(1)}} (\Delta_{i\xi}^{(1)} - \mu_i) \prod_{\gamma=i+1}^{\alpha} \prod_{\xi=1}^{\beta^{(1)}} (\Delta_{\gamma\xi}^{(1)} - \mu_\gamma) \leq \prod_{\xi=j+1}^{\beta^{(1)}} \Delta_{i\xi}^{(1)} \prod_{\gamma=i+1}^{\alpha} \prod_{\xi=1}^{\beta^{(1)}} \Delta_{\gamma\xi}^{(1)},$$

то приходим к выражению:  $Dist \leq Z^{(1)}$ .

При этом код числа  $Dist$  в раздельном пространстве, определяемый как относительное расстояние, будет меньше, чем разность между кодом числа  $Z^{(1)}$  и кодом нижнего уровня  $Z_{\min}^{(1)}$  в исходном квантованном пространстве, т.е.

$$Dist \leq Z^{(1)} - Z_{\min}^{(1)}. \quad (1)$$

Истинность выражения (1) доказывает тот факт, что использование представления массива элементов верхнего квантованного уровня в раздельном пространстве приводит к устранению комбинаторной избыточности, которое достигается:

– уменьшением разброса значений элементов верхнего квантованного уровня путем выявления минимальных значений, т.е. выполнения неравенств

$$(x_{ij}^{(1)} - v_i) \leq x_{ij}^{(1)} \text{ и } (\Delta_{ij}^{(1)} - v_i) \leq \Delta_{ij}^{(1)} ;$$

– сокращением количества чисел, для которых не соблюдается условие дифференциального пространства и которые предшествуют текущему числу.

Суммарное количество  $\text{Numb}^{(2)}$  позиционных чисел, для которых выполняется условие в дифференциальном неравновесном пространстве, рассчитывается согласно следующему выражению:

$$\text{Numb}^{(2)} = \prod_1^{\alpha} \prod_{j=1}^{\beta^{(1)}} (\Delta_{ij}^{(1)} - v_i) . \quad (2)$$

Формула (2) указывает на сокращенную комбинаторную избыточность в результате применения представления массива верхнего квантованного уровня в раздельном пространстве. Это приводит к уменьшению временных затрат на доведение сильнонасыщенных фрагментов изображения.

### 3. Выводы

1) Разработан подход к формированию квантованного описания трансформанты сегмента аэрофотоснимка на основе двухуровневого представления динамического диапазона значений трансформанты, что позволяет сократить комбинаторную избыточность.

2) Обосновано, что использование представления массива элементов верхнего квантованного уровня в раздельном пространстве приводит к дополнительному сокращению комбинаторной избыточности, вследствие чего уменьшается время на доведение данных аэрофотоснимка.

3) Разработана модель оценки количества сокращенной комбинаторной избыточности на основе представления массива верхнего квантованного уровня в раздельном пространстве.

**Список литературы:** 1. Баранник В.В., Підлісний С.А. Аналіз дії кібератак на відео інформаційний ресурс в інформаційно-телекомунікаційних мережах // АСУ та прилади автоматики. 2014. Вип. 164. С. 16-22. 2. Баранник В.В. Кодирование трансформированных изображений в инфокоммуникационных системах / В.В. Баранник, В.П. Поляков. Х.: ХУПС, 2010. 212 с. 3. Олифер В.Г. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. / В.Г. Олифер, Н.А. Олифер. СПб.: Питер, 2006. 958 с. 4. Красильников Н.Н. Цифровая обработка изображений. М.: Вузовская книга, 2011. 320 с. 5. Баранник В.В. Метод сжатия изображений на основе неравновесного позиционного кодирования битовых плоскостей / В.В. Баранник, Н.К. Гулак, Н.А. Королева // Радіоелектронні і комп'ютерні системи. Х.: ХНАУ "ХАІ", 2009. Вип. 1. С. 55– 61. 6. Баранник В.В. Кодирование трансформированных изображений в инфокоммуникационных системах / В.В. Баранник, В.П. Поляков. Х.: ХУПС, 2010. 212 с.

Поступила в редколлегию 11.12.2015

**Баранник Дмитрий Владимирович**, студент факультета КИУ ХНУРЭ. Научные интересы: кодирование и защита информации для передачи в телекоммуникационных системах. Адрес: Украина, 61166, Харьков, пр. Науки, 14, e-mail: barannik\_v\_v@mail.ru.

**Стасев Сергей Юрьевич**, аспирант ХНУРЭ. Научные интересы: кодирование и защита информации для передачи в телекоммуникационных системах. Адрес: Украина, 61000, Харьков, ул. Сумская, 77/79. E-mail: barannik\_v\_v@mail.ru.

**Медведев Денис Олегович**, аспирант ХНУРЭ. Научные интересы: кодирование и защита информации для передачи в телекоммуникационных системах. Адрес: Украина, 61000, Харьков, ул. Сумская, 77/79. E-mail: barannik\_v\_v@mail.ru.

## АНАЛИЗ ХАРАКТЕРИСТИК СИСТЕМ CDMA ОТНОСИТЕЛЬНО ИНТЕНСИВНОСТИ БИТОВОГО ПОТОКА

Рассматривается эволюция систем сотовой связи. Анализируются переходные этапы развития от поколения к поколению. Описывается метод кодового разделения независимых каналов связи, а также рост битового потока при использовании CDMA. Вычисляется коэффициент избыточности битового потока.

### Введение

Современный мир пользователей (сотовой связи, интернета и т.д.) нуждается в более высокой скорости обмена данными и увеличении количества одновременно обслуживаемых пользователей в общем канале связи [3].

В этом направлении технологии развиваются уже несколько десятилетий [2]. Увеличение скорости обмена данными происходит за счет:

- 1) расширения количества используемых параметров;
- 2) частоты (первое поколение) [3];
- 3) частоты, интервалов времени (второе поколение) [3];
- 4) частоты, интервалов времени, кода Уолша - энергетический параметр сигнала (третье поколение) [1], [3];
- 5) привязки точки приема и точки излучения к конкретным координатам на одной плоскости ( система MIMO - Multiple Input Multiple Output) [3];
- 6) увеличения спектральной ширины сигналов обмена данными [1];
- 7) организации каналов связи на более высокой частоте ( свыше 10 МГц) [1].

Увеличение количества одновременно обслуживаемых абонентов в общем эфире и параллельно во времени осуществляется за счет:

- дробления обслуживаемой территории на более мелкие ячейки в целях уменьшения возможного количества пользователей в зонах повышенной интенсивности применения трафика [3];
- использования ортогональных последовательностей (коды Уолша) [1];
- применение ортогональных частот (позволяет эффективно использовать частотный диапазон канала связи) [1].

*Цель работы* – проанализировать существующие методы передачи данных, выявить и исследовать недостатки существующих систем, рассчитать коэффициент битовой избыточности.

*Актуальность* данной работы заключается в анализе двух основных проблем:

- скорость передачи данных;
- индивидуальная анонимность информационного сообщения.

### 1. Анализ эволюции систем сотовой связи

История развития связи имеет огромное количество гениальных идей усовершенствования и конструктивных решений. Первым методом разделения нескольких абонентов в общем эфире было частотное разделение доступа (FDM) Frequency - Division Multiplexing, что позволило в одно время без препятствий организовать работу больше одного канала связи. В связи с повышением количества любителей воспользоваться данными изобретения частотного разрешенного пространства стало недостаточно. Тогда было разработано и реализовано временное распределение доступа (TDM) Time Division Multiplexing, что позволило эффективнее использовать частотно-временной ресурс. Количество абонентов продолжало увеличиваться, и следствием стала разработка кодового распределения доступа (CDMA) Code Division Multiple Access, которая позволила говорить уже о частотно-временном-энергетическом распределении каналов связи. CDMA технология в разы увеличила количество абонентов и улучшила качество связи.

CDMA основан на использовании ансамбля 64-х ортогональных кодов Уолша, которые в свою очередь кодируют 1024 ортогональных поднесущих модуляции (OFDM) Orthogonal frequency-division multiplexing. Несколько поднесущих и их результат OFDM – сигнала можно посмотреть на рис.1.

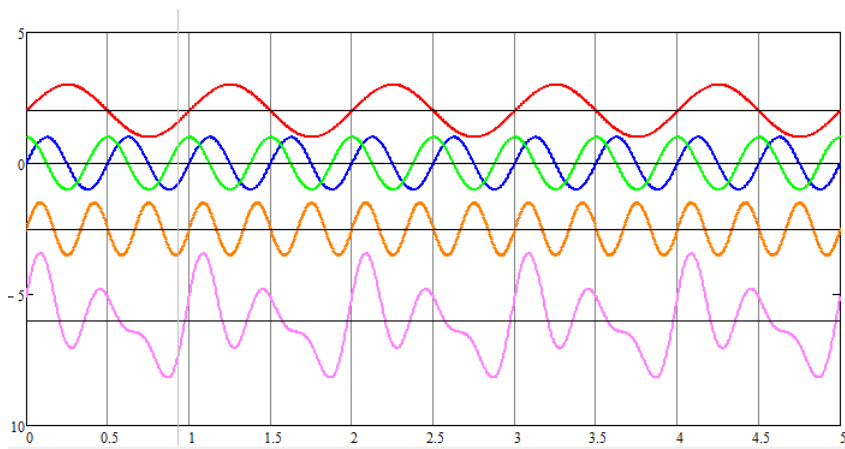


Рис. 1. Совокупность поднесущих сигналов и их результат сложения при формировании OFDM - сигнала

Эта конструкция позволяет в компактном спектре частот параллельно во времени организовать несколько каналов связи без взаимных препятствий.

## 2. Свойства кодового разделения доступа, расчет избыточности битового потока

Для организации CDMA используют коды Уолша, обладающие свойствами полной ортогональности, что позволяет без взаимных помех организовать несколько независимых каналов связи для разных абонентов на единственной частоте параллельно во времени. Количество каналов связи прямо пропорционально зависит от длины кодовой последовательности.

Увеличение количества возможных независимых каналов связи на единственной частоте приводит к увеличению длины кодирующей последовательности Уолша, которая в свою очередь уменьшает скорость передачи информации по каналу связи. Зависимость скорости трафика от количества независимых каналов достаточно простая и имеет вид:

$$R_{\text{реал}} = \frac{R_{\text{факт}}}{N_{\text{кан}}}, \quad (1)$$

где  $R_{\text{факт}}$  – фактическая (максимальная) скорость канала связи без использования распределения информационного пространства между разными абонентами;  $N_{\text{кан}}$  – количество организованных независимых каналов связи на единственной несущей частоте параллельно во времени.

Использование всех независимых каналов не обязательно. В случае необходимости возможно увеличивать или уменьшать количество абонентов в общем эфире.

Рассматривая частичную нагрузку абонентами единого пространства передачи данных с ортогональным разграничением, наблюдаем зависимость помехоустойчивости системы от количества абонентов. Помехоустойчивость системы эквивалентна отношению:

$$E_{\text{сист}} \approx \frac{(N_{\text{max}})^2}{N_{\text{реал}}}, \quad (2)$$

где  $N_{\text{max}}$  – максимально возможное количество независимых каналов связи;  $N_{\text{реал}}$  – реальное количество абонентов, которые использует данный канал одновременно.

Коды Уолша строятся несколькими методами [1].

Метод Радемахера превращает синусоиду согласно алгоритму:

$$r_n(x) = \begin{cases} +1, & \text{если } \sin(n \cdot x) > 0; \\ -1, & \text{если } \sin(n \cdot x) < 0. \end{cases} \quad (3)$$

Полученная последовательность имеет вид рис. 2.

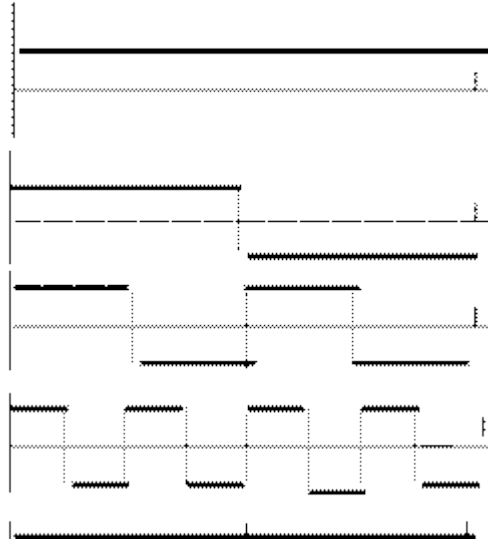


Рис. 2. Ансамбль функций Уолша, полученных методом Радемахера

Для получения функций Уолша из системы функций Радемахера используют преобразование:

$$\text{wal}(w, : \Theta) = \prod_{k=1}^n [r_k(\Theta)]^{w_{n-k+1} \oplus w_{n-k}}, \quad (4)$$

где  $r_k$  – функция Радемахера;  $k$  – номер соответствующей функции;  $\text{wal}(w, : \Theta)$  – функции Уолша на основе функций Радемахера;  $w$  – порядок соответствующей функции Уолша.

Для получения функций Уолша, упорядоченных по Пели, используют преобразование:

$$\text{pal}(p, : \Theta) = \prod_{k=1}^n [r_k(\Theta)]^{p_{n-k+1}}. \quad (5)$$

Метод Адамара применяет матрицу Адамара для формирования последовательности:

$$H_2 = \begin{vmatrix} H_1 & H_1 \\ H_1 & -H_1 \end{vmatrix} = \begin{vmatrix} 1 & 1 \\ 1 & -1 \end{vmatrix}, \quad (6)$$

$$H_4 = \begin{vmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{vmatrix} = \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{vmatrix}. \quad (7)$$

Если детально рассмотреть приведенные выше методы формирования ортогональных кодовых последовательностей, то оказывается, что состав ансамблей одинаков и отличается лишь упорядочением. Наглядная совокупность этих кодов показана на рис. 3.

Набор ортогональных функций упорядочен по  $w$  (порядковый номер функции Уолша), далее поставлен в соответствие  $p$  (номер функций Уолша, упорядоченных по Пели),  $h$  (номер функции Уолша, сформированной на базе матриц Адамара).

Ортогональные коды хоть и обладают высочайшими свойствами помехоустойчивости, но также они чрезмерно нагружают канал связи избыточностью.

Поскольку формирование ортогональных кодов для всех систем одинаково и отличается лишь порядком следования при одинаковой форме сигнала, то для расшифровки достаточно использовать простой метод перебора. Если учесть, что этот перебор состоит из 64 взаимоисключающих вариантов, то нужный код будет найден моментально. Данный анализ показывает, что ортогональные коды не обладают криптостойкостью.

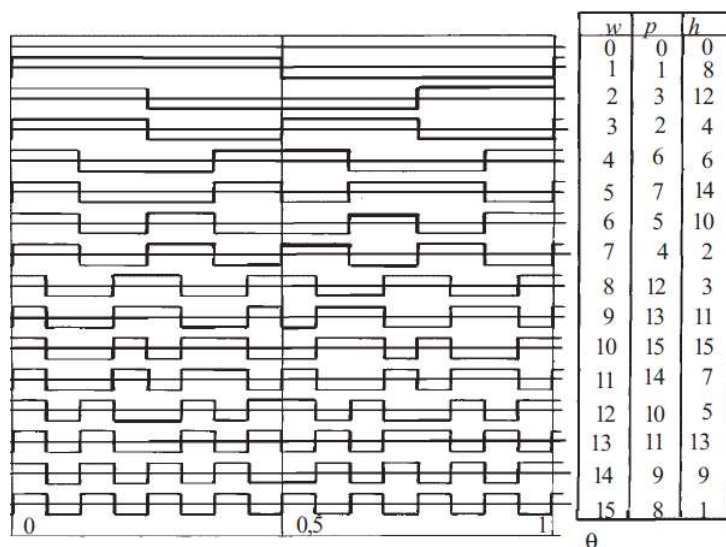


Рис. 3. Ансамбль функций Уолша и их нумерация по Пели и Адамару

### 3. Описание и наглядный пример формирования ортогональных кодов

Битовый поток в вычислительных системах существует в виде последовательности положительных или отрицательных напряжений, 0 соответствует напряжению условное: -1, а 1 условное: +1 (табл.1).

Информационная последовательность  $In_N^q$ , где q – номер абонента (кода Уолша),  $N = n \cdot 1, 1 = 64$  – длина кодовой последовательности Уолша, перемножается на ортогональный код Уолша  $wal(q)_n$  длина которого помещается на интервале следования информационного бита:

$$S_n^q = In_N^q \cdot wal(q)_n. \quad (8)$$

Результатом является ортогональное сообщение:  $S_n^q$  (8). Далее совокупность ортогональных сообщений попадает в общий канал, где побитно суммируется. Сумма общего потока  $Sm_n$  передается в эфир:

$$Sm_n = \sum_{q=0}^{q_{\max}} S_n^q. \quad (9)$$

На приемной стороне (табл. 2,3) для извлечения своей информации из общего суммарного потока каждый абонент использует свою ортогональную, кодовую последовательность Уолша (соответствующую кодовой последовательности на передающей стороне) для извлечения информационного сообщения, адресованного ему лично.

Таблица 1

Ортогональное разделение																		
$q = 1$																		
$\text{In}_N^1$	0	0	0	1	0	1	1	1	0	0	0	1	0	1	1	1	0	0
$\text{wal}(1)_n$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$S_n^1$	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
$q = 2$																		
$\text{In}_N^2$	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
$\text{wal}(2)_n$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$S_n^2$	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
$q = 3$																		
$\text{In}_N^3$	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\text{wal}(3)_n$	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1
$S_n^3$	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1
$q = 4$																		
$\text{In}_N^4$	0	1	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
$\text{wal}(4)_n$	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1
$S_n^4$	-1	1	-1	1	-1	1	-1	-1	1	-1	1	-1	1	-1	1	-1	-1	1
Сложение нескольких каналов в один поток																		
$S_n^1$	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
$S_n^2$	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1
$S_n^3$	1	-1	1	-1	-1	1	-1	1	1	-1	1	-1	-1	1	-1	1	1	-1
$S_n^4$	-1	1	1	-1	-1	-1	1	-1	1	1	-1	-1	1	-1	-1	1	1	-1
$S_{m_n}$	-2	-2	2	-2	-2	-2	2	-2	2	-2	-2	2	2	2	0	0	-4	0

Таблица 2

Приемный канал первого абонента (изъятие информации из общего ортогонально сложенного потока данных)																													
$S_{m_n}$	-2	-4	2	0	-1	-2	-2	2	-2	-4	2	0	-2	0	-2	4	0	0	0	0	4	0	0	0	4	0	0	0	4
$\text{wal}(1)_n$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$S_{1n}^1$	-2	-4	2	0	-2	-2	-2	2	-2	-2	2	0	-2	2	2	2	0	0	-4	0	0	0	0	4	0	0	0	4	0
$S_{2n}^1$	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
$\text{In}_N^1$	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Таблица 3

Приемный канал второго абонента (изъятие информации из общего ортогонально сложенного потока данных)																														
Sm <sub>n</sub>	-2	-2	2	-2	-2	-2	-2	2	2	2	-2	-2	2	2	2	0	0	-4	0	0	0	0	4	0	0	0	0	4	0	4
wal(2) <sub>n</sub>	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	-1	-1
S1 <sub>n</sub> <sup>2</sup>	-2	-2	-2	2	-2	-2	2	-2	-2	2	-2	-2	2	-2	0	2	-4	0	0	0	0	4	0	0	0	0	0	0	0	-4
S2 <sub>n</sub> <sup>2</sup>			-4			-4			-4			-4			-4			4			-4			-4			-4		-4	
In <sub>N</sub> <sup>2</sup>			-1			-1			-1			1			1			1			1			1			1		1	
			0			0			0			0			0			1			0			0			0		0	

Для этого необходимо перемножить побитно весь поток на свою кодовую последовательность:

$$S1_n^q = Sm_n \cdot wal(q)_n \cdot \quad (10)$$

Далее берется сумма всех результатов в пределах длины последовательности Уолша:

$$S2_N^q = \sum_{n=1}^1 S1_n^q, \quad (11)$$

после чего восстанавливается информационное (цифровое) сообщение:

$$In_N^q = \frac{S2_N^q}{1}. \quad (12)$$

#### 4. Расчет избыточности битового потока в CDMA

Цифровая информация  $In_N^q$  перед передачей в эфир проходит несколько этапов обработки, увеличивая свою избыточность  $W$  на выходе  $Sm_n$  [1]:

– 1 бит информации сверочным кодированием преобразовывается в 2 бита (повышение помехоустойчивости);  $In_N^q \cdot 2$ ;

– перемежение (равномерное распределение информации по всему временному интервалу - для борьбы с «замиранием»)  $In_N^q \cdot 2 \cdot 1$ ;

– скремблирование (позволяет защитить канал связи от несанкционированного доступа)  $In_N^q \cdot 2 \cdot 1 \cdot 1$ ;

– преобразование ортогональным кодом Уолша порядка  $l = 64$  (позволяет организовать 64 независимых канала связи в совместной среде распространения)  $In_N^q \cdot 2 \cdot 1 \cdot 1 \cdot 1$ ;

– сложение ансамбля кодовых сообщений  $\sum_{q=1}^{q_{\max}} In_N^q \cdot 2 \cdot 1 \cdot 1 \cdot 1$ ;

– далее следует манипуляция аналогового сигнала по закону цифрового ряда сформированной кодовой последовательности на основе функции Уолша (понятие бита исчезает);

Зная алгоритм преобразований в CDMA, рассчитаем битовую избыточность для максимально возможного количества абонентов  $q = 64$ :

$$W = \frac{\sum_{q=1}^{q_{\max}} In_N^q \cdot 2 \cdot 1 \cdot 1 \cdot 1}{In_N^q} = \frac{64 \cdot (In_N^q \cdot 2 \cdot 64)}{In_N^q} = 8192. \quad (13)$$

Рассчитанная битовая избыточность в современных системах связи имеет довольно большое значение.

## Заключение

Анализ современных систем связи показал:

1. Современная система CDMA имеет избыточность 8192 бита на выходе к 1 биту информации, что приводит к чрезмерной загруженности канала связи и снижению скорости обмена данными. В свою очередь, такой факт является неблагоприятным и требует решения для обеспечения потребностей абонентов по увеличению скорости обмена данными.

2. Отсутствует криптостойкость ортогонально кодированных сообщений и неспособность обеспечения необходимого уровня конфиденциальности. Данное обстоятельство выходит за рамки требований международного права о сохранении личной информации и противоречит тенденции растущих запросов безопасности и конфиденциальности обмена данными среди потребителей.

3. Ограничение количества независимых каналов связи с общим ресурсом в связи со свойством ортогонального кода: для увеличения независимых каналов связи необходимо увеличивать длину ортогональной последовательности. Из формулы (13) видно, что избыточность будет возрастать по квадратному закону.

**Список литературы:** 1. *Никитин Г.И.* Применение функций Уолша в сотовых системах связи с кодовым разделением каналов // Редакционно-издательский отдел электронных публикаций и библиографии библиотеки. Отдел оперативной полиграфии СПбГУАП 190000, Санкт-Петербург. 2. *Аверин В.Г.* Компьютерные сети и телекоммуникации. Курс лекций. Федеральное государственное образовательное учреждение среднего профессионального образования «Уральский радиотехнический техникум им. А.С. Попова», Екатеринбург 2009. 3. *Диязитдинов Р.Р.* Системы подвижной связи. Конспект лекций. Самара: ФГОБУ ВПО ПГУТИ, 2013.

*Поступила в редколлегию 11.12.2015*

**Баранник Владимир Викторович**, д-р техн. наук, профессор, начальник кафедры Харьковского университета Воздушных Сил. Научные интересы: кодирование и защита информации для передачи в телекоммуникационных системах. Адрес: Украина, 61023, Харьков, ул. Сумская, 77/79. E-mail: barannik\_v\_v@mail.ru

**Окладной Дмитрий Евгеньевич**, начальник группы (авиационного прицельно - навигационного комплекса) учебно-лабораторного комплекса кафедры бортового оборудования и комплексов воздушной разведки инженерно - авиационного факультета Харьковского университета Воздушных Сил. Адрес: Украина, 61023, Харьков, ул. Сумская, 77/79, E-mail: barannik\_v\_v@mail.ru

**Супрун Олег Валерьевич**, соискатель кафедры Информационно-сетевой инженерии ХНУРЭ. Научные интересы: кодирование и защита информации для передачи в телекоммуникационных системах. Адрес: Украина, 61166, Харьков, пр. Науки, 14.