ФЕДЕРАЛЬНОЕ АГЕНТСТВО НАУЧНЫХ ОРГАНИЗАЦИЙ

ВСЕРОССИЙСКИЙ ИНСТИТУТ НАУЧНОЙ И ТЕХНИЧЕСКОЙ ИНФОРМАЦИИ
РОССИЙСКОЙ АКАДЕМИИ НАУК
(ВИНИТИ РАН)

НАУЧНО • ТЕХНИЧЕСКАЯ ИНФОРМАЦИЯ

Серия 2. ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ И СИСТЕМЫ

ЕЖЕМЕСЯЧНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ СБОРНИК

Издается с 1961 г. № 6 Москва 2019

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

УДК 004.89:510.64

С.М. Гусакова, А.Н. Охлупина

Интеллектуальная ДСМ-система как средство автоматизированной поддержки научных исследований в почерковедении

Описана интеллектуальная система поддержки научных исследований в почерковедении. ДСМ-метод, разработанный для этой системы, моделирует рассуждения эксперта и максимально учитывает особенности решаемых задач

Ключевые слова: *почерковедение, идентификация подписи, операция сходства, ДСМ-метод для атрибутов с весами, правила правдоподобного вывода*

ВВЕДЕНИЕ

Необходимость компьютерной поддержки работы эксперта-почерковеда признана всеми в криминалистическом сообществе. Однако вопрос о видах этой поддержки вызывает разногласия. Наиболее востребованы программные продукты, позволяющие уменьшить трудоемкость и субъективизм при выделении экспертом признаков почерка [1]. К системам, решающим экспертные задачи (идентификационные и диагностические), отношение неоднозначное. Автоматические системы (см., например, [2]) не используются в судебном почерковедении, так как суд принимает только обоснованный вывод эксперта, которому необходимо понимать, что делает система, а её действия должны производиться в привычных для эксперта категориях. В связи с отмечаемой специалистами тенденцией к упрощению подписи и, следовательно, к уменьшению ее информативности [3],

возникает вопрос о необходимости использования новых, более объективных методов в почерковедении.

Еще один вид компьютерной поддержки экспертно-почерковедческой деятельности, который пока не рассматривается и разработки в этом направлении отсутствуют – это интеллектуальные системы. Одной из таких систем, которая может быть использована для этой деятельности, является ДСМ-система автоматизированной поддержки научных исследований (ДСМ-АПНИ). Различные методы, входящие в системы типа ДСМ, реализуют синтез познавательных процедур – индукции, аналогии и абдуктивного принятия полученных в процессе работы системы гипотез. Модель предметной области, позволяющая создать формализованный язык представления данных и определить содержательную операцию сходства для нахождения гипотез о причинах свойств изучаемых объектов, а также база фактов, содержащая примеры (факты), отражающие сведения о наличии и отсутствии исследуемых свойств объектов, являются основой для применения различных стратегий ДСМ-метода, позволяющих извлекать новые знания, имплицитно содержащиеся в базе фактов (подробно о ДСМ-методе и интеллектуальных ДСМ-системах автоматизированной поддержки научных исследований см. в [4, 5]).

Как вариант решения задачи идентификации подписи в работе [6] был предложен модифицированный ДСМ-метод [7], использующий в качестве операции сходства объектов, представленных в виде множеств, операцию объединения. Этот выбор обусловлен успешным применением указанного метода для решения задач, концептуально близких решаемой [8, 9].

Анализ результатов проведенных экспериментов показал, что идентификация подписи в виде определения ее исполнителя при сравнении с множеством подлинных и поддельных образцов имеет особенности, не полностью учитываемые модифицированным ДСМ-методом. Кроме того, актуальная задача создания средств автоматизированной поддержки научных исследований в почерковедении, в первую очередь связанных с выделяемыми признаками и их значимостью, не может быть решена с помощью модифицированного ДСМ-метода. Поэтому был разработан новый вариант ДСМ-метода, учитывающий особенности решаемых задач, моделирующий рассуждения эксперта и позволяющий проводить исследовательскую работу в автоматизированном режиме.

При разработке этого нового варианта ДСМ-метода для решения поставленных задач максимально учитывались знания и интуиция эксперта в области почерковедения.

ОСОБЕННОСТИ ЯЗЫКА ПРЕДСТАВЛЕНИЯ ДАННЫХ

База фактов ДСМ-системы содержит положительные примеры – образцы подлинных подписей и их собственноручно выполненных расшифровок, а также отрицательные примеры – образцы поддельных подписей и расшифровок. Все они записаны в языке представления данных, включающем транскрипцию, общие и частные признаки почерка. Транскрипция описывает общий вид подписи, список букв и их порядок и безбуквенных элементов в ней. Общие и частные признаки собраны в списки и пронумерованы. Общие признаки состоят из имени признака и из его значения, а их нумерация включает две цифры: *I.j*, где *I* – номер имени общего признака, *j* – номер его значения.

Частные признаки устроены сложнее. Они делятся на группы, в каждой из которых признак может относиться к одной или двум буквам или к безбуквенным элементам. В букве различаются элементы, в элементах могут быть выделены части, называемые конкретизацией элемента (верхняя точка, нижняя точка и т.п.). В безбуквенных элементах также возможно выделение частей. Все это составляет частный признак. Каждый признак имеет несколько значений, которые иногда могут быть уточнены, т.е. здесь тоже имеется конкретизация.

Для исследовательской составляющей работы с системой, а также для решения других задач с использованием этой же базы фактов (что предполагается в дальнейшем) может понадобиться определение сходства для частных признаков на разных уровнях – уровне группы, имени признака или значения в имени, поэтому нумерация частных признаков учитывает все эти уровни и имеет вид: $I; j{:}k_1, k_2; m_1, m_2$. Здесь I – номер группы, j – номер буквы (пары букв) или сама буква (пара букв), k_1 – элемент буквы, k_2 – конкретизация элемента буквы, m_1 – значение признака, m_2 – конкретизация значения. Конкретизации элемента буквы и значения могут отсутствовать. В этом случае k_2 и/или m_2 равно 0.

Такое разбиение частных признаков по иерархическому принципу учитывается в интерфейсе подсистемы ввода, предлагающем эксперту при выборе или вводе нового признака выбрать группу, затем букву или безбуквенный элемент или их пары; после чего – элемент буквы, если нужно, конкретизацию элемента буквы и, наконец, выбрать или ввести значение признака, возможно с конкретизацией. Этот принцип позволяет эксперту не набирать длинные названия признаков и поддерживать единообразие в их описании.

ОСОБЕННОСТИ РЕШАЕМЫХ ЗАДАЧ

Предлагаемый в настоящей работе вариант ДСМ-метода предназначен как для решения ряда почерковедческих задач, так и для проведения исследований в почерковедении.

Опишем использование интеллектуальной системы, основанной на варианте ДСМ-метода, разработанного для решения задачи идентификации и исследовательской работы, связанной с определением значимости признаков и групп признаков.

При выборе варианта ДСМ-метода необходимо учитывать ряд особенностей задач идентификации подписи и определения значимости ее признаков и групп признаков:

- задача идентификации подписи решается каждый раз для одного человека, при этом информация о подписях других лиц не используется. Это значит, что база фактов разбивается на подбазы и задача решается в каждой такой подбазе. Для исследований, связанных с признаками, привлекается и материал,

полученный в каждой подбазе, и обобщающий материал по всем наборам подписей разных людей;

- в связи со свойствами устойчивости и вариативности почерка, а также с разной частотностью встречаемости признаков и их значений признаки подписи неравнозначны;
- свойство вариативности почерка требует учитывать тот факт, что признак с его значением, встретившийся среди всех образцов подписи одного и того же человека только один раз, тоже должен приниматься во внимание при формировании индуктивного вывода;
- при принятии решения о подлинности или фальсификации исследуемой подписи эксперт рассматривает каждый признак в отдельности и только потом оценивает всю совокупность выделенных признаков;
- поскольку в базе фактов есть подлинные и поддельные образцы подписей, то сама подделка приводит к довольно точному воспроизведению многих признаков подлинных подписей. Это означает, что многие совпадающие в подлинных и поддельных подписях признаки имеют одинаковое значение;
- при выводе эксперта о подлинности или подделности исследуемого образца подписи, им оцениваются аргументы «за» и «против», причем среди и тех и других есть неравнозначные;
- аргументами «за» или «против» являются признаки не только входящие в исследуемый образец, но и не входящие в него, причем последние более весомы, чем первые.
- вывод эксперта имеет форму категорического или вероятного «за» или «против» подлинности исследуемого образца, или формулируется как «не представляется возможным сделать вывод»;
- если признак, встретившийся одновременно в образцах подлинных и поддельных подписей с одинаковым значением, часто повторяется от человека к человеку, это свидетельствует о том, что его легко подделать;
- частные признаки, выделяемые экспертами при описании образцов подписей, распределяются по группам неравномерно с достаточно устойчивой частотой встречаемости от эксперта к эксперту.

Все эти указанные особенности являются аксиомами предметной области для описанных нами задач. Поэтому они были учтены при выборе варианта ДСМ-метода, стратегии ДСМ-рассуждений и при формулировках правил правдоподобного вывода.

ВАРИАНТ ДСМ-МЕТОДА ДЛЯ РЕШЕНИЯ ЗАДАЧ КРИМИНАЛИСТИЧЕСКОГО ИССЛЕДОВАНИЯ ПОДПИСИ

Для решения поставленных задач в первую очередь формируется база фактов (БФ), которая состоит из положительных ($БФ^+$) и отрицательных ($БФ^-$) примеров. Каждый положительный пример – это один из образцов подписи конкретного человека и ее расшифровки, записанный в языке представления данных. Отрицательные примеры – аналогичные образцы, но выполненные другим человеком от лица первого. Таким образом, пример в $БФ^+$ имеет вид: $J_{(1,n)}\ (X \Rightarrow_1 W)$, где 1 указывает, что пример положительный, n – номер шага рассуждения (для базы фактов $n = 0$) и читается как «образец подписи с расшифровкой (объект) X выполнен лицом W». Объект X записывается в виде: $X= \{(I;\ j{:}k_1,\ k_2;\ m_1,\ m_2)_f\}$, $f = 1,\ldots F$, где F – число признаков с учетом различных значений, выделенных экспертом в данном образце.

Пример в $БФ^-$ имеет вид $J_{(-1,n)}\ (X \Rightarrow_1 W)$ и читается как «образец подписи X выполнен другим лицом от имени лица W».

Неравнозначность признаков требует введения для них весов, а это определяет выбор варианта ДСМ-метода для атрибутов (признаков) с весами [10] для решения задачи идентификации подписи. В качестве весов логично было бы взять идентификационные значимости признаков почерка (логарифмы величин, обратных к частоте встречаемости признака в почерках разных людей). Но они во многом устарели, так как почерки и подписи современных людей изменились по сравнению с почерками предыдущих поколений [11], а для безбуквенных элементов, очень часто встречающихся в подписях, эти значимости не подсчитывались. Поэтому в качестве веса частного признака была взята степень его устойчивости в образцах подписи и расшифровках, т.е. $p(q_i)$ – вес признака q со значением i равен количеству образцов, в которых такой признак с этим значением встретился в совокупности всех образцов подписи данного человека. Эта характеристика существенна для оценки значимости признака.

Для решения задачи идентификации выбрана однородная стратегия $\{M^+_{a,n}\ (V, W),\ \ M^-_{a,n}\ (V, W)\}$, т.е. в качестве предикатов сходства взяты предикаты простого положительного и отрицательного сходства [12] с некоторыми изменениями, учитывающими особенности задачи, указанные ранее. Предикаты сходства участвуют в формулировке правил индуктивного вывода (п.п.в.-1), которые порождают гипотезы 1-го рода о причинах выполнения или невыполнения свойств объектов. Гипотезам присваивается одно из истинностных значений $\{+1,-1,0,\tau\}$, указывающих, является ли подобъект V положительной (+1), отрицательной (-1) или противоречивой (0) гипотезой о причине выполнения свойств. Если истинностное значение равно τ, то сохраняется неопределенность. Эти истинностные значения определяются логическим соотношением выполнения или невыполнения предикатов $M^+_{a,n}\ (V, W)$ *и* $M^-_{a,n}\ (V, W)$.

Гипотезы 1-го рода записываются в виде:
$J_{(v,n)}\ (V \Rightarrow_2 W)$, $v \in \{+1,-1,0,\tau\}$, где n – номер шага, на котором получена гипотеза.

Подобъект V является: положительной гипотезой, т.е. причиной выполнения свойства W, если выполняется $M^+_{a,n}\ (V, W)$ и не выполняется $M^-_{a,n}\ (V, W)$; отрицательной гипотезой – в противоположном случае; противоречивой гипотезой, если выполняются оба предиката. В случае, если ни один из них не выполняется для V, гипотеза получает истинностное значение τ.

Особенности задачи потребовали внесения некоторых корректировок в формулировку предикатов сходства.

Во-первых, операция сходства объектов из базы фактов определяется на каждом признаке отдельно,

т.е. $X \sqcap Y = \{X/q \sqcap Y/q \mid X/q,\ Y/q$ – проекции объектов $X,\ Y$ на признак с именем $q\}$. Для общих признаков $X/q \sqcap Y/q = (I,j)^x \sqcap (I,j)^y = (I,j)^x$, если $(I,j)^x = (I,j)^y$ или λ, если $(I,j)^x \neq (I,j)^y$. Здесь λ – пустой элемент.

Для частных признаков в зависимости от постановки задачи могут быть определены три операции сходства:

$$X/q \sqcap_1 Y/q = (I;\ j{:}k_1,\ k_2;\ m_1,\ m_2)^x \sqcap_1 (I;\ j{:}k_1,\ k_2;\ m_1,\ m_2)^y = (I;\ j{:}k_1,k_2;m_1,m_2)^x,$$

если

$$(I;j{:}k_1,k_2;m_1,m_2)^x = (I;\ j{:}k_1,k_2;m_1,m_2)^y. \quad (1)$$

$$X/q \sqcap_2 Y/q = (I;\ j{:}k_1,\ k_2;\ m_1,\ m_2)^x \sqcap_2 (I;\ j{:}k_1,\ k_2;\ m_1,\ m_2)^y = I^x,$$

если

$$I^x = I^y \quad (2)$$

$$X/q \sqcap_3 Y/q = (I;\ j{:}k_1,\ k_2;\ m_1,\ m_2)^x \sqcap_3 (I;\ j{:}k_1,\ k_2;\ m_1,\ m_2)^y = (I;\ j{:}k_1,k_2)^x,$$

если

$$(I;\ j{:}k_1,k_2)^x = (I;\ j{:}\ k_1,\ k_2)^y. \quad (3)$$

Для всех трех операций результат равен λ, если равенство не имеет места.

Во-вторых, условие $k \geq 2$, где k – минимальное количество примеров, которые могут породить гипотезу 1-го рода, меняется на $k \geq 1$.

Таким образом, предикат $M^+_{a,n}\ (V,\ W)$ приобретает вид:

$$M^+_{a1,n}\ (V_q,\ W) = \exists k\ \exists X_1 \ldots \exists X_k\ (\&^k_{j=1}\ (J_{(1,n)}(X_j \Rightarrow_1 W)) \ \&\ ((X_1/q \sqcap \ldots \sqcap X_k/q) = V_q) \ \&\ (V_q \neq \varnothing) \ \&\ \forall a\ \forall b\ (((a \neq b) \ \&\ (1 \leq a,\ b \leq k)) \rightarrow X_a \neq X_b)) \ \&\ \forall X\ (J_{(1,n)}(X \Rightarrow_1 W) \ \&\ (V_q \subset X)) \rightarrow \& \ (\bigvee_{i=1}^{k} (X = X_i)) \ \&\ (k \geq 1).$$

Если гипотеза $(V_q,\ W)$ не удовлетворяет предикату $M^-_{a1,n}\ (V_q,\ W)$, который определяется аналогично с заменой положительных примеров на отрицательные, то она получает истинностное значение +1. Если имеет место $M^-_{a1,n}\ (V_q,\ W)$ и не имеет места $M^+_{a1,n}\ (V_q,\ W)$ – гипотеза получает истинностное значение -1. В случае, если гипотеза $(V_q,\ W)$ удовлетворяет и предикату $M^+_{a1,n}\ (V_q,\ W)$ и $M^-_{a1,n}\ (V_q,\ W)$, она получает истинностное значение 0.

Правила вывода по аналогии (п.п.в.-2) в классическом ДСМ-методе дают возможность доопределить объекты, свойства которых не известны. Объект определяется как обладающий исследуемым свойством, если он включает положительные гипотезы и не включает отрицательные и противоречивые. Наличие в объекте отрицательных гипотез и отсутствие положительных и противоречивых позволяет говорить о нем, как о не обладающем исследуемым свойством. В случае наличия и положительных, и отрицательных гипотез и/или противоречивых – доопределение противоречиво. Объект, не содержащий ни одной гипотезы, остается недоопределенным.

При решении вопроса идентификации подписи правила вывода по аналогии требуют уточнений, отражающих особенности этой задачи. Проводя экспертное исследование, криминалист обращает внимание как на признаки, совпадающие у образца подписи, представленной на экспертизу, и образцов подписей предполагаемого исполнителя, так и на различающиеся признаки, которые даже более значимы. Из этого следует, что правила вывода по аналогии должны проверять не только какие положительные и отрицательные гипотезы входят в исследуемый образец, но и какие гипотезы, как положительные, так и отрицательные не входят в него. В результате такой проверки для каждого предъявленного на экспертизу образца T получается набор, состоящий из четырех множеств: $\{V^+, V^-, Z^+, Z^-\}$, где $V^+ = \{V^+(q_i)\}$ – множество положительных гипотез, входящих в T; $V^- = \{V^-(r_j)\}$ – множество отрицательных гипотез, входящих в T; $Z^+ = \{Z^+(t_l)\}$ – множество положительных гипотез, не входящих в T; $Z^- = \{Z^-(s_m)\}$ – множество отрицательных гипотез, не входящих в T; $q,\ r,\ t,\ s$ – имена признаков; $i,\ j,\ l,\ m$ – значения признаков.

Поскольку для решения задачи был выбран вариант ДСМ-метода для атрибутов с весами, каждому признаку, входящему в гипотезу, должен быть приписан вес, который определяется следующим образом: признак, проявившийся в образцах подписи человека, только один раз получает вес 1; для признаков, встретившихся от 2 до N раз, где N – количество всех положительных или отрицательных примеров для одного человека, вес определяется с помощью разделения числового отрезка $[2,\ N]$ равномерно на n отрезков. Каждому из них присваивается число по порядку: от двух – для самого левого отрезка, до n – для самого правого. Гипотеза получает вес, равный номеру отрезка, в который входит число образцов, содержащих данную гипотезу. Для положительных гипотез этот вес имеет знак плюс (+), для отрицательных – знак минус (-).

С учетом того, что гипотезы, не вошедшие в исследуемый образец, тоже учитываются и их значимость для вывода больше, чем у вошедших гипотез, веса этих гипотез умножаются на коэффициент 1,5. Причем, вес положительных гипотез имеет знак минус, а отрицательных – знак плюс. Смена знаков весов на противоположные объясняется тем, что если положительная гипотеза (т.е. значение признака, который встречается в положительных примерах и не встречается в отрицательных) не входит в предъявленный на экспертизу образец, то это является достаточно сильным аргументом против того, что образец подлинный (аналогично с отрицательной гипотезой).

Положительные и отрицательные веса складываются и получается два числа: P^+ и P^-. Следует учесть, что вхождение и положительных, и отрицательных гипотез в исследуемый образец не приводит в общем случае к противоречивому выводу. Вывод определяется сравнением разности суммарных весов положительных и отрицательных гипотез с заданными порогами.

Противоречивые гипотезы не проверяются на вхождение в исследуемый образец. Это связано с тем, что противоречивые гипотезы всегда входят в образцы независимо от того, подлинными или поддельными они являются, поскольку специфика подделки подписи определяет большое количество противоречивых гипотез. Признаки, входящие в эти гипотезы, неинформативны для решения вопроса идентификации подписи конкретного человека и могут быть удалены на уровне препроцессинга данных.

Поскольку в исследуемый образец подписи могут входить как положительные, так и отрицательные гипотезы, разность суммарных весов P^+-| P^-| сравнивается с одним из двух порогов μ_1 и μ_2, причем $\mu_1 > \mu_2$. Если $P^+ \neq 0$ и $P^- \neq 0$, то используется порог μ_1, если $P^+ = 0$ или $P^- = 0$ – то используется меньший порог μ_2, потому что отсутствие гипотез противоположного знака в исследуемом образце тоже аргумент «за».

Вывод оценивается следующими истинностными значениями: {+1,-1, +1/2, -1/2, 0, τ}.

Таким образом, предикаты, выражающие процедуру вывода по аналогии, приобретают вид:

$$\Pi_n^{+1}(X,W) \rightleftharpoons$$

$$\exists k1, k2, k3, k4\ \exists V_q ((\&_{q=1}^{k1} (J_{(1,n)} (V_q \Rightarrow_2 W)) \&$$

$$(V_q \subset X)) \& \exists Z_t (\&_{t=1}^{k3} (J_{(1,n)} (Z_t \Rightarrow_2 W)) \& \neg (Z_t \subset X)) \&$$

$$\exists V_r (\&_{r=1}^{k2} (J_{(-1,n)} (V_r \Rightarrow_2 W)) \& (V_r \subset X))$$

$$\& \exists Z_s (\&_{s=1}^{k4} (J_{(-1,n)} (Z_s \Rightarrow_2 W)) \& \neg (Z_s \subset X)) \&$$

$$(\Sigma_{q=1}^{k1} V_q + \Sigma_{s=1}^{k4} Z_s = P^+) \&$$

$$(\Sigma_{r=1}^{k2} V_r + \Sigma_{t=1}^{k3} Z_t = P^-) \& (((P^+ \neq 0 \ \& \ P^- \neq 0) \rightarrow$$

$$(P^+ - |P^-|) > \mu_1) \vee ((P^+ = 0 \vee P^- = 0) \rightarrow (P^+ - |P^-|) > \mu_2))).$$

$$\Pi_n^{+1/2}(X,W) \rightleftharpoons$$

$$\exists k1, k2, k3, k4, \exists V_q ((\&_{q=1}^{k1} (J_{(1,n)} (V_q \Rightarrow_2 W)) \&$$

$$(V_q \subset X)) \& \exists Z_t (\&_{t=1}^{k3} (J_{(1,n)} (Z_t \Rightarrow_2 W)) \& \neg (Z_t \subset X)) \&$$

$$\exists V_r (\&_{r=1}^{k2} (J_{(-1,n)} (V_r \Rightarrow_2 W)) \& (V_r \subset X)) \&$$

$$\exists Z_s ((\&_{s=1}^{k4} (J_{(-1,n)} (Z_s \Rightarrow_2 W)) \& \neg (Z_s \subset X)) \&$$

$$((\Sigma_{q=1}^{k1} V_q + \Sigma_{s=1}^{k4} Z_s = P^+) \&$$

$$(\Sigma_{r=1}^{k2} V_r + \Sigma_{t=1}^{k3} Z_t = P^-) \& (((P^+ \neq 0 \ \& \ P^- \neq 0) \rightarrow$$

$$(P^+ - |P^-|) = \mu_1) \vee ((P^+ = 0 \vee P^- = 0) \rightarrow (P^+ - |P^-|) = \mu_2))).$$

$$\Pi_n^{-1}(X,W) \rightleftharpoons$$

$$\exists k1, k2, k3, k4, \exists V_q ((\&_{q=1}^{k1} (J_{(1,n)} (V_q \Rightarrow_2 W)) \&$$

$$(V_q \subset X)) \& \exists Z_t (\&_{t=1}^{k3} (J_{(1,n)} (Z_t \Rightarrow_2 W)) \&$$

$$\neg (Z_t \subset X)) \& \exists V_r (\&_{r=1}^{k2} J_{(-1,n)} (V_r \Rightarrow_2 W)) \& (V_r \subset X)) \&$$

$$\exists Z_s ((\&_{s=1}^{k4} (J_{(-1,n)} (Z_s \Rightarrow_2 W)) \& \neg (Z_s \subset X)) \&$$

$$((\Sigma_{q=1}^{k1} V_q + \Sigma_{s=1}^{k4} Z_s = P^+) \& ((\Sigma_{r=1}^{k2} V_r + \Sigma_{t=1}^{k3} Z_t) =$$

$$P^-) \& (((P^+ \neq 0 \ \& \ P^- \neq 0) \rightarrow (|P^-| - P^+) > \mu_1) \vee$$

$$((P^+ = 0 \vee P^- = 0) \rightarrow (|P^-| - P^+) > \mu_2))).$$

$$\Pi_n^{-1/2}(X,W) \rightleftharpoons$$

$$\exists k1, k2, k3, k4, \exists V_q ((\&_{q=1}^{k1} (J_{(1,n)} (V_q \Rightarrow_2 W)) \&$$

$$(V_q \subset X)) \& \exists Z_t (\&_{t=1}^{k3} (J_{(1,n)} (Z_t \Rightarrow_2 W)) \& \neg (Z_t \subset X)) \&$$

$$\exists V_r (\&_{r=1}^{k2} (J_{(-1,n)} (V_r \Rightarrow_2 W)) \& (V_r \subset X)) \&$$

$$\exists Z_s ((\&_{s=1}^{k4} (J_{(-1,n)} (Z_s \Rightarrow_2 W)) \& \neg (Z_s \subset X)) \&$$

$$((\Sigma_{q=1}^{k1} V_q + \Sigma_{s=1}^{k4} Z_s) = P^+) \& ((\Sigma_{r=1}^{k2} V_r + \Sigma_{t=1}^{k3} Z_t) =$$

$$P^-) \& (((P^+ \neq 0 \ \& \ P^- \neq 0) \rightarrow (|P^-| - P^+) < \mu_1) \vee$$

$$((P^+ = 0 \vee P^- = 0) \rightarrow (|P^-| - P^+) < \mu_2))).$$

$$\Pi_n^{\tau}(X,W) \rightleftharpoons \forall V_q ((J_{(1,n)} (V_q \Rightarrow_2 W)) \rightarrow$$

$$\neg (V_q \subset X)) \& \forall V_r (J_{(-1,n)} (V_r \Rightarrow_2 W)) \rightarrow \neg (V_r \subset X)).$$

Если выполняется предикат $\Pi_n^{+1}(X,W)$, то делается «категорический вывод», что подпись X подлинная; для предиката $\Pi_n^{-1}(X,W)$ – «категорический вывод», что подпись X поддельная; выполнение предикатов $\Pi_n^{+1/2}(X,W)$ и $\Pi_n^{-1/2}(X,W)$ позволяет сделать вероятный вывод, что подпись X подлинная или поддельная соответственно; если выполнен предикат $\Pi_n^{0}(X,W)$ – сделать вывод не представляется возможным; значение τ в предикате вывода по аналогии означает, что тестируемый образец остался недоопределнным. Это значение присваивается в случае отсутствия вхождения как положительных, так и отрицательных гипотез в тестируемый образец и указывает эксперту на необходимость уточнить описание образца исследуемой подписи, и, быть может, образцов подписей в базе фактов.

Отметим, что принятая в криминалистике формулировка о категорическом положительном или отрицательном выводе, за исключением нескольких случаев, не является с логической точки зрения достоверным выводом, а только правдоподобным. Достоверный – это, например, категорический отрицательный вывод, когда признак «степень выработанности почерка» в представленном на идентификацию образце подписи имеет значение «ниже средней», а для образцов подписей с которыми она сравнивается – значение «выше средней», так как повысить выработанность невозможно. Однако субъективизм эксперта при оценке степени выработанности может поставить под сомнение достоверность и в этом случае.

Для правдоподобных выводов по аналогии, полученных с помощью ДСМ-системы, можно утверждать, что степень правдоподобия тем выше, чем больше разность суммарных весов положительных и отрицательных гипотез.

Выбор весов и порогов носит предварительный характер. Окончательный выбор должен быть сделан после проведения серии экспериментов.

ЭКСПЕРИМЕНТАЛЬНАЯ ПРОВЕРКА РАБОТЫ СИСТЕМЫ

Для проведения эксперимента у тридцати лиц (курсанты, слушатели, адъюнкты и преподаватели Московского университета МВД России имени В.Я. Кикотя, сотрудники следственных, оперативных и экспертных подразделений органов внутренних дел МВД России, а также иные лица) были собраны образцы подписей и собственноручно выполненных расшифровок этих подписей.

Экспериментальный материал собирался единообразно – на линованном бланке участники эксперимента выполняли подписи и их расшифровки. Бланк содержал также данные об исполнителе – фамилию, имя, отчество, дату рождения и должность. Чтобы свести к минимуму появления признаков, вызванных непривычными условиями, все испытуемые заполняли бланки в обычной обстановке: сидя за письменным столом, при хорошем освящении, шариковой ручкой, в привычном темпе. В результате были собраны 30 бланков с 11 образцами подписей и расшифровками на каждом из них.

Эти образцы передали иным лицам с целью подделки на таких же бланках при аналогичных условиях с предварительной тренировкой. Бланки содержали по 10 поддельных подписей с расшифровками.

Все подписи с расшифровками с помощью подсистемы ввода были введены в базу фактов в разработанном языке представления данных. Из каждого бланка и с подлинными, и с поддельными подписями были выбраны по одному образцу для дальнейшей идентификации. Оставшиеся образцы подлинных подписей составили положительную часть базы фактов ($БФ^+$), а образцы поддельных – ее отрицательную часть ($БФ^-$).

Таким образом, экспериментальная база фактов содержала 300 положительных и 270 отрицательных примеров. На идентификацию были представлены тридцать подлинных и тридцать поддельных образцов подписей. Поскольку для каждого человека исследование проводилось отдельно, по существу $БФ^+$ и $БФ^-$ были разбиты на тридцать подбаз, в каждой из которых на идентификацию представлялось два образца – подлинный и поддельный.

Сначала идентификация проводилась без учета расшифровок. В ходе эксперимента по идентификации исполнителей подписей использовались как общие, так и частные признаки. Эксперимент дал следующие результаты: из тридцати неопределенных в эксперименте, но реально подлинных подписей с оценкой +1 (категорически подлинная) была идентифицирована 21 подпись; из тридцати неопределенных в эксперименте, но реально поддельных подписей с оценкой -1 (категорически неподлинная) была идентифицирована 21 подпись, две подлинные и две поддельные подписи доопределились как вероятно подлинные и вероятно поддельные с оценкой +1/2 и -1/2 соответственно, четыре подлинных и четыре поддельных подписей – с оценкой 0 в форме «не представляется возможным сделать вывод» (*далее* – НПВ), одна подлинная подпись определилась неверно как поддельная, две поддельные подписи определились неверно как подлинные, и три (две подлинных и одна неподлинная) остались недоопреленными.

Для 18-ти подписей с оценками, отличными от +1 и -1, был проведен дополнительный эксперимент с использованием почерковой информации, содержащейся в расшифровках исполнителей исследуемых подписей. В результате добавилось 14 правильных доопределений. Из оставшихся четырех случаев, которые не получили истинностных оценок +1 или -1, образцы двух подписей доопределены правильно, но в вероятной форме, а еще для двух – вывод сделан в форме НПВ. Существенно, что для этих четырех образцов расшифровки не были введены в базу фактов. Их экспертная оценка показала, что объекты, определившиеся в вероятной форме, состоят из монограммы и короткого росчерка, подделка подписи выполнена на очень хорошем уровне, причем все признаки являются устойчивыми во всех образцах. В этой ситуации эксперт сделал бы вывод в форме НПВ, в то время как ДСМ-система позволила определить эти объекты правильно, хотя в вероятной форме.

В двух случаях, когда система дала ответ в форме «не представляется возможным сделать вывод», установлено, что подпись выполнена простыми движениями и информации для более определенного вывода недостаточно.

Для оценки эффективности работы описываемой системы исследуемые объекты были предложены эксперту-почерковеду на идентификацию с помощью качественно-описательной методики. Для 22-х из 30-ти подлинных подписей экспертом был дан ответ в форме НПВ (следует отметить, что ответы в форме НПВ эксперты дают примерно в 30% случаев [13]).

Результаты проведенного эксперимента позволяют сделать вывод, что выбранный для решения задачи идентификации исполнителя подписи вариант ДСМ-метода достаточно эффективен. В случаях, когда система выдает неудовлетворительные результаты, эффективность её работы может быть увеличена за счет привлечения дополнительной почерковой информации, полученной из расшифровок.

АВТОМАТИЗИРОВАННАЯ ПОДДЕРЖКА НАУЧНЫХ ИССЛЕДОВАНИЙ В ПОЧЕРКОВЕДЕНИИ

Созданная интеллектуальная ДСМ-система позволяет не только решать задачу идентификации. Она также является средством автоматизированной поддержки научных исследований в почерковедении.

Как было упомянуто ранее, противоречивые гипотезы в правилах вывода по аналогии не участвуют. Однако они служат материалом для изучения диагностической значимости признаков. Для такого исследования была привлечена вся база фактов по всем людям, оставившим как подлинные, так и поддельные подписи. С этой целью каждый признак, входящий в противоречивую гипотезу для одного человека, был проверен на вхождение в противоречивые гипотезы для всех людей, чьи образцы подписей внесены в базу фактов.

В результате проверки выявлены три признака, значения которых повторялись в противоречивых гипотезах для достаточного количества людей из базы фактов, что свидетельствует о том, что такие признаки легко воспроизводятся при подделке. Все три признака относятся к безбуквенным элементам – росчерку и штриху. Вывод о малой информативности этих признаков для безбуквенных элементов, сделанный благодаря использованию системы, особенно важен с учетом того, что идентификационная значимость безбуквенных элементов в почерковедении не подсчитывалась. Небольшое количество выделенных малоинформативных элементов объясняется тем, что в подписях разных людей мало одинаковых букв и элементов. С увеличением количества положительных и отрицательных примеров в базе фактов число этих элементов будет увеличиваться. Выявленные элементы, не несущие информации для экспертного вывода, могут быть исключены из списка признаков или получить соответствующую пометку в нем, что должно резко уменьшить вес этого признака в случае попадания его в положительную или отрицательную гипотезу.

Другое исследование, которое проводилось с помощью ДСМ-системы, относится к распределению частных признаков, вошедших в противоречивые и положительные гипотезы, по группам частных признаков. Для этого исследования также использовалась вся база фактов. Эксперимент, описанный в [14], показал, что наибольшее количество частных признаков, выделенных разными экспертами на одном и том же материале, попадает в группу «относительное размещение движений при выполнении букв и элементов», далее идет группа – «форма движений при выполнении букв и элементов». Число признаков, выделенных в этих двух группах, превышает 50% от общего числа выделенных частных признаков.

Сравнение распределения по группам частных признаков, вошедших в противоречивые и положительные гипотезы, с устойчивым распределением всех выделенных экспертом частных признаков позволяет выявить более или менее информативные для идентификации группы частных признаков.

Распределение по группам всех частных признаков, а также признаков, вошедших в противоречивые и положительные гипотезы, в абсолютном и процентном соотношении относительно всех выделенных признаков приведены в таблице.

Из сравнения распределений по группам признаков, входящих в гипотезы, с распределением признаков всей базы фактов по группам видно, что в целом эти распределения близки, но есть некоторые отклонения. Так, количество признаков из положительных гипотез в процентном отношении больше количества признаков всей базы для группы №2 «Форма движений при выполнении букв и элементов», а для группы №3 «Форма движений при выполнении соединения букв и элементов» – меньше. Количество признаков из этой же группы, входящих в противоречивые гипотезы, напротив, отклоняется в сторону увеличения. Это указывает на то, что признаки из группы №3 «Форма движений при выполнении соединения букв и элементов» легче поддаются подделке, чем из группы №2 «Форма движений при выполнении букв и элементов». Следовательно, эти признаки менее информативны для идентификации подписи и должны иметь меньший вес по сравнению с признаками из других групп.

Распределение по группам признаков почерка

№	Группа признаков	Количество частных признаков, А.Ц./%		
		Все частные признаки в группах	Частные признаки из противоречивых гипотез	Частные признаки из положительных гипотез
1	Относительное размещение движений при выполнении букв и элементов	405/41,7%	174/40,1%	59/39,1%
2	Форма движений при выполнении букв и элементов	194/19,2%	90/20,8%	41/27,2%
3	Форма движений при выполнении соединений букв и элементов	75/7,7%	43/9,9%	5/3,3%
4	Протяженность движений по вертикали при выполнении букв и элементов	104/10,7%	44/10,1%	17/11,3%
5	Протяженность движений по горизонтали при выполнении букв и элементов	78/8,0%	25/5,8%	12/7,9%
6	Сложность движений при выполнении букв и элементов	10/1,0%	4/0,9%	2/1,3%
7	Количество движений при выполнении букв и элементов	42/4,3%	16/3,7%	6/4,0%
8	Направление движений при выполнении букв и элементов	33/3,4%	22/5,1%	4/2,6%
9	Вид соединения при выполнении букв и элементов	28/2,9%	16/3,7%	4/2,6%
10	Угол между элементами	2/0,2%	0/0%	1/0,7%
Всего признаков в группе		**971**	**434**	**151**

Конечно, объем массива данных недостаточен для статистически значимых выводов. Однако необходимо учитывать, что выявление информативности признаков в автоматизированном режиме в почерковедении не проводилось, а проблема изучения информативности в целом группы признаков вообще не ставилась. Поэтому наличие методики проведения таких исследований с помощью ДСМ-системы, которая позволит при расширении массива данных делать более обоснованные выводы, открывает новые возможности для работы криминалиста-почерковеда.

С помощью описанной системы можно проводить и другие виды исследований (см., например, [15]).

ЗАКЛЮЧЕНИЕ

Если учесть, что идентификация исполнителя подписи имеет дело с объектом, при реализации которого «процесс автоматизированного письма почти превращается в простую “моторную идеограмму”» [16], становится понятно, какая это сложная задача. При проведении судебно-почерковедческой экспертизы эксперт должен обосновать сделанный вывод. Автоматические системы, решающие задачу идентификации подписи, такого обоснования не дают, поэтому не используются при проведении судебно-почерковедческих экспертиз. Интеллектуальная ДСМ-система автоматизированной поддержки научных исследований моделирует рассуждения эксперта и обосновывает сделанный вывод, поэтому она может быть использована для работы эксперта-почерковеда. А возможность в автоматизированном режиме проводить исследования и выявлять свойства признаков почерка открывает перспективы для проведения исследований в области почерковедения на новом уровне.

СПИСОК ЛИТЕРАТУРЫ

1. Смирнов А.В. Программа «ОКО-1» для исследования кратких и простых почерковых объектов // Теория и практика судебной экспертизы. – М.: ГУ РФЦСЭ, 2006. – Вып. 1. – С. 121-124.
2. Sheng He, Schomaker L. Writer identification using curvature-free features // Pattern Recognition. – 2017. – Vol. 63. – P. 451-464.
3. Вилкова Н.А., Кошманов М.П., Кошманов П.М. «Служебный» вариант подписи – гарантия ее как полноценного реквизита документа // Эксперт-криминалист. –2014. – №3. – С. 21-25.
4. Финн В.К. Индуктивные методы Д.С. Милля в системах искусственного интеллекта. Часть I // Искусственный интеллект и принятие решений. – 2010. – № 3. – С. 3-21.
5. Финн В.К. Индуктивные методы Д.С. Милля в системах искусственного интеллекта. Часть II // Искусственный интеллект и принятие решений. – 2010. – № 4. – С. 14-40.
6. Гусакова С.М., Лапшина И.А., Охлупина А.Н. Идентификация подписи: постановка задачи и вариант решения с помощью интеллектуальной ДСМ-системы // Научно-техническая информация. Сер. 2. – 2018 – №8. – С. 8-13.
7. Гусакова С.М. Подход к решению задач атрибуции исторических источников с помощью ДСМ-метода // Автоматическое порождение гипотез в интеллектуальных системах / под ред. проф. В.К. Финна. – М.: Книжный дом «Либроком», 2009. – С. 494-501.
8. Гусакова С.М., Комаров А.С. Интеллектуальная система для решения идентификационной задачи в почерковедении // Искусственный интеллект и принятие решений. – 2010. – №4. – С. 49-54.
9. Устинов В.В. Модельные методы судебно-почерковедческого исследования: дис. … канд. юридич. н. – М.: Моск. ун-т МВД РФ им. В. Я. Кикотя, 2011.
10. Липкин А.А. ДСМ-метод порождения гипотез для объектов, описываемых атрибутами с весами: дис. … канд. техн. наук. – М., 2008. – 117 с.
11. Сысоева Л.А. Современное состояние почерковедческого исследования подписи // Фотография. Изображение. Документ. – 2013. – № 4(4). – С. 10-14.
12. Финн В.К. Эпистемологические основания ДСМ-метода автоматического порождения гипотез // Научно-техническая информация. Сер. 2. Часть I. – 2013. – № 9. – С. 1-29; Часть II. – № 12. – С. 1-26; Finn V.K. Epistemic foundations of the JSM method automatic hypothesis generation // Automatic Documentation and Mathematical Linguistics. – 2014. – Vol. 48, №2. – P. 96-148.
13. Кошманов П.М., Кошманов М.П, Шнайдер А.А. Удостоверительная защитная функция подписи // Нотариус. – 2010. – Вып. 3. – С. 40-45.
14. Охлупина А.Н. Проблема однозначности выделения признаков подписей и ее влияние на процесс автоматизации экспертных исследований // Алтайский юридический вестник. – 2019. – №1(26). – С. 115-120.
15. Гусакова С.М. Зависимость особенностей подписи от психофизиологических характеристик ее исполнителя: подход к решению задачи // 16 национальная конференция по искусственному интеллекту с международным участием (КИИ-18), 24-27 сентября 2018, Вороново, Москва - Труды конференции - том 1 – 2018 - С.198-204.
16. Лурия А.Р. Очерки психофизиологии письма – М.: Академия пед. наук РСФСР, 1950. – 84 с.

Материал поступил в редакцию 22.03.19

Сведения об авторах

ГУСАКОВА Светлана Марковна – кандидат физико-математических наук, старший научный сотрудник Федерального исследовательского центра «Информатика и управление» РАН, Москва
e-mail: svem45@yandex.ru

ОХЛУПИНА Анастасия Николаевна – адъюнкт Московского университета МВД РФ им. В.Я. Кикотя, Москва
e-mail: stasya.zharova@inbox.ru

УДК 002.6:004.056.5

А. А. Грушо, М.И. Забежайло, В.О. Писковский, В.В. Сенчило, Е.Е.Тимонина

Подходы к интеграции информационных систем цифровой экономики*

Описаны два подхода к интеграции информационных систем – MCPS и DCPS. Проанализированы их преимущества и недостатки. Представлен обзор открытых проектов в классе решений «Очередь сообщений» (Message Queue – MQ). Основное внимание уделено новому, активно развивающемуся направлению – интеграции информационных систем в облачных вычислительных средах в целях развития информационной инфраструктуры цифровой экономики. Обсуждается как текущее состояние, так и перспективные возможности использования подходов при решении задач информационной безопасности.

Ключевые слова: *Цифровая экономика, информационная системы, ACID, OpenStack, MCPS, DCPS, MOM, MQ, DDS, SLA, QoS, MDM, Serializability Theorem, мультиверсионность*

ВВЕДЕНИЕ

Одна из основных особенностей цифровой экономики – взаимодействие разнородных информационных систем, функционирующих в традиционной экономике. Переход к цифровой экономике от традиционной представляется в основном за счёт автоматизации взаимодействия существующих систем при соблюдении принципов невмешательства в их работу и абсолютной прозрачности и управляемости, вплоть до юридической значимости безусловно фиксируемых событий межсистемного взаимодействия.

Типичные современные информационные системы – распределенные и системы, реализующие функционирование экономики, – не являются исключением. Под распределенностью понимается как типичная многозвенная архитектура современных систем (приложений), так и интеграция нескольких автономных сервисов, взаимодействующих между собой.

Ключевым аспектом того, что распределённые системы могут существовать и выполнять свою роль, является обмен информацией. Основной вопрос при этом: как доставить точную информацию от ее источников к потребителям за время, не превышающее заранее заданного порога. Поскольку выбор способа доставки влияет на каждое приложение, он определяет многие ключевые свойства взаимодействия разнородных информационных систем, включая своевременность, масштабируемость, надежность, доступность, конфигурацию и развитие.

В распределенных системах, по сравнению с централизованными, многие понятия, такие как целостность, транзакции, состояния, приобретают более сложное комплексное структурное значение. Одна из важнейших функций распределённых систем цифровой экономики – корректное управление комплексными структурными понятиями в соответствии с принятыми требованиями:

- *ACID* (*Atomicity* – Атомарность, *Consistency* – Согласованность, *Isolation* – Изолированность, *Durability* – Долговечность);
- совместного управления и синхронизации состояний замкнутых систем, взаимодействующих в рамках Цифровой экономики.

Ранее в других статьях мы касались проблем, возникающих при работе распределённых систем в облачных вычислительных средах. Одна из них – управление реконфигурацией программно-конфигурируемых сетей посредством введения транзакций и аномалий на уровень управления сетевыми устройствами – была рассмотрена в работе [1]. Приведенные в ней рассуждения и выводы применимы и к взаимодействующим системам, если рассматривать каждую из таких систем, как атомарный обработчик данных. Еще одна проблема – управление и синхронизация состояний – обсуждалась в работах [1,2], в том числе на примере работы комплекса проектов *OpenStack*. Ниже будут рассмотрены методы, позволяющие подойти к решению проблем, поднятых в этих статьях, и составляющих единую задачу управления распределенной инфраструктурой взаимодействующих систем.

* Работа выполнена при поддержке РФФИ (проект 18-29-03081 офи-м – ориентированные фундаментальные исследования по актуальным междисциплинарным темам)

ДВА ПОДХОДА К ИНТЕГРАЦИИ

Существенную роль в решении поставленной задачи играет подход к интеграции. На сегодняшний день можно выделить два подхода: 1) ориентирующийся на доставку сообщений – *Message-Centric Publish-Subscribe* (*MCPS*), 2) направленный на обмен унифицированными данными – *Data-Centric Publish-Subscribe* (*DCPS*). Оба эти подхода реализуют обмен сообщениями между приложениями. Происходит это благодаря программному обеспечению, специально созданному для решения таких задач. Назовём его системой обмена сообщениями или программным обеспечением интеграционного уровня (*integration layer* или *middleware*).

Подход MCPS

В зависимости от выбранного подхода различается роль системы обмена сообщениями. В *MCPS* форматно-логический контроль, синтаксический и семантический анализ осуществляются на уровне приложений.

Технология *MCPS* обеспечивает высокоскоростную асинхронную межпрограммную связь с надёжной доставкой. Программы обмениваются пакетами данных, называемыми сообщениями. Каналы или очереди являются логическими путями, которые соединяют программы и передают сообщения. Канал ведёт себя как набор или массив сообщений, используется несколькими компьютерами или приложениями одновременно. Отправитель (*Writer*) – программа, которая отправляет сообщение, записывает его в канал. Получатель или подписчик – программа, которая получает сообщение, читает и при необходимости, удаляет его из канала.

Само сообщение является структурой данных (строка, байтовый массив, запись или объект). Его можно интерпретировать как данные, описание команды, которая должна быть вызвана на получателе, описание события, произошедшего в отправителе и т.д. Сообщение состоит из двух частей: заголовка и тела. Заголовок включает метаинформацию о сообщении – кто его отправил, куда он направляется и т.д.; эти сведения используются системой обмена сообщениями и, в основном (но не всегда), игнорируются использующими их приложениями. Тело содержит передаваемые данные и игнорируется системой обмена сообщениями [3].

Возможности такого обмена сообщениями обычно предоставляются специальной системой или промежуточным программным обеспечением – *message-oriented middleware (MOM*) [5], координирующим и управляющим отправкой и получением информации. Необходимость выделить систему класса *MOM* вызвана, с одной стороны, отсутствием условий синхронизации между приложениями, участвующими в обмене сообщений: если приложение *А* готово отправить сообщение, это не означает, что приложение *Б*, для которого оно предназначено, именно в этот момент готово его принять, так как действие, вызванное приёмом этого сообщения, может нарушить корректную работу приложения *Б*. С другой стороны, сети и компьютеры в сети – суть автономные системы, которые не могут и не должны постоянно быть доступны для обмена информацией хотя бы по причине необходимости проведения обязательных регламентных и технологических работ по замене, обновлению, расширению оборудования и программного обеспечения.

Система обмена сообщениями – отдельная автономная информационная система, задача которой изолировать связываемые приложения, обеспечивать обмен информацией между ними и системами независимо от их состояния, повышая тем самым «ремонтопригодность» и надежность распределенной системы в целом. Система обмена сообщениями принимает сообщения, если необходимо, накапливает их и передает адресату в соответствии с принятым соглашением об уровне обслуживания – *Service Level Agreement* (*SLA*), обеспечивая принятое для распределённой системы качество обслуживания – *Quality of Service (QoS*).

На рис. 1 показана схема работы системы обмена сообщениями на примере системы *IBM MQ* [6].

На диаграмме (рис. 2) представлены принципиальные возможности горизонтальной масштабируемости и обеспечения надёжности распределённой системы за счёт резервирования и использования кластера ресурсов [7,8].

Завершая обзор методов работы *MCPS*, также укажем на существование двух подходов к обмену сообщениями. В рамках первого агент системы обмена не хранит на стороне отправителя в памяти или на диске оригинал отправляемого сообщения, а обеспечивает его хранение и доставку своими силами. В рамках второго подхода агент, прежде чем отправить сообщение, сохраняет его на стороне отправителя, а после завершения отправки агент, размещенный на стороне получателя, также сохраняет у себя полученные данные при их сложной маршрутизации, например, пересылке через посредника. Этот процесс повторяется несколько раз, пока данные не достигнут получателя.

Отметим, что в рамках первого подхода встречаются случаи, когда нет смысла гарантировать стопроцентную доставку сообщений при упрощении системы в пользу достижения высоких скоростей обмена. Этот тип доставки применяется в работе распределенных систем, где ставится задача массового оперативного информирования клиентов, а не соблюдение принципа *ACID*, который мы рассмотрели выше. Область применения этих систем: торговые площадки, биржи, аукционы, где значимость факта потери сравнительно небольшого процента сообщений ничтожно мала по сравнению со сколь-нибудь значимой задержкой информации об изменившейся ситуации на рынке, например, цене на лот.

Помимо описанного сценария доставки от одного отправителя к одному получателю возможно, что один входной канал разделяется на несколько выходных, по одному для каждого абонента, на которые пересылается одна и та же информация. Каждый выходной канал имеет только одного подписчика, получающего сообщение только один раз. При этом доставленные копии исчезают из своих каналов. Особенность такой групповой рассылки – необходимость наличия отдельного механизма для подтверждения отправителю факта доставки информации

получателю или получения ответа. Для этого необходимы два канала: 1)канал, по которому отправитель отсылает групповое сообщение и ожидает ответ от каждого получателя;2) канал, по которому получатель отправляет ответное сообщение.

Для синхронизации отправки запроса и получение на него ответа есть два подхода: синхронный и асинхронный. Выбор между ними основан на анализе всего графа процессов, выполняемых распределённой системой, с целью построения допустимого непротиворечивого (консистентного) плана по аналогии с реконфигурацией облачных вычислительных сред, или применения теоремы «Упорядочиваемости» (*Serializability Theorem*) [1]. Как показывает анализ, избежать при этом аномалий [1-3] вряд ли удастся, но возможно предусмотреть риски их возникновения в соответствии с выбранным уровнем сериализации, а также разработать способы компенсации в каждом конкретном случае. Отметим, что процессы и системы в цифровой экономике в основном относятся к системам реального времени. Поэтому метод мультиверсионности, когда для сохранения целостности изменения в данных накапливаются параллельно с актуальными неизменяемыми данными и вводятся в действие при наличии определённых условий, имеет естественные ограничения для использования.

Перечислим преимущества системы обмена сообщениями [9,10]:

- обеспечение «ремонтопригодности» и масштабируемости распределённой системы;
- поддержка версионности входящих в распределенную систему приложений и обеспечения развития;
- мониторинг и управление работой распределённой системы, что немаловажно для ее информационной безопасности;
- обеспечение связи для логически и географически распределенной системы;
- мультиплатформенность, возможность использовать приложения, написанные на разных платформах с использованием разных технологий;
- поддержка выполнения допустимого непротиворечивого (консистентного) плана выполнения процедур с возможностью минимизации рисков возникновения аномалий;
- обеспечение высокой доступности, балансировки нагрузки, выполнения заданных параметров качества услуг и уровня обслуживания за счёт использования резервных ресурсов.

К недостаткам системы обмена сообщениями можно отнести:

- сложность модели программирования, использование в общем случае событийно-ориентированной модели;
- необходимость тщательного планирования и отработки всего процесса функционирования распределённой системы, а также анализа условий изоляции при построении корректного плана;
- потерю производительности как расплаты за удобство и универсальность.

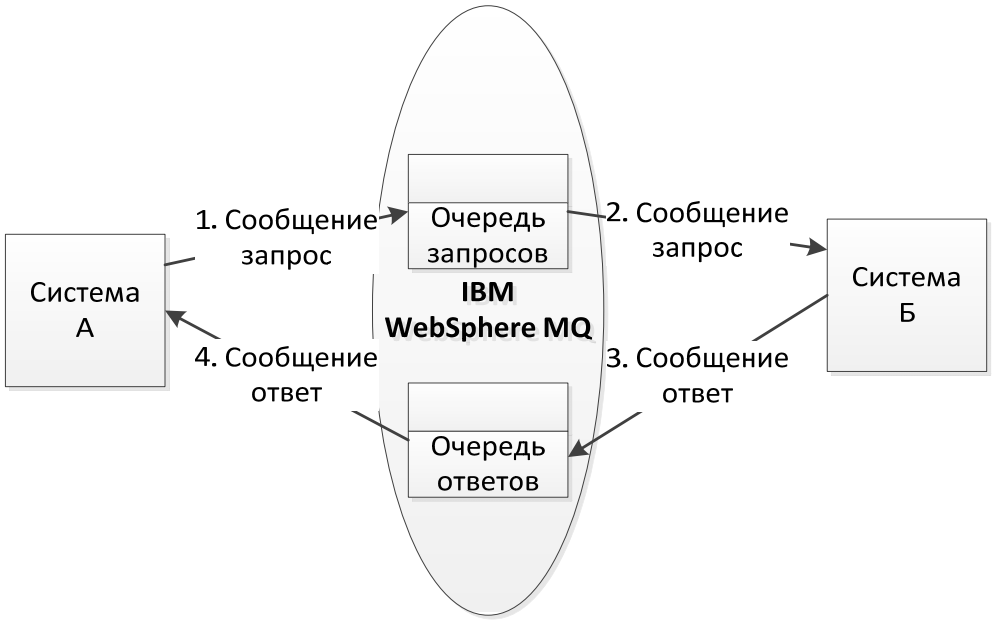


Рис. 1. Схема работы системы *IBM MQ*

где: 1 – система *А* отправляет запрос системе *Б*; 2 – промежуточный слой в виде *IBM MQ* сохраняет запрос в своей очереди и в соответствии с бизнес-логикой и доступностью системы *Б* переправляет запрос; 3 – ответ от системы *Б*, даже если это просто ответ об успешном получении запроса, проходит обратно такой же путь.

Необходимо отметить, что многие распространённые проприетарные (являющиеся собственностью авторов или правообладателей и не удовлетворяющее критериям свободного программного обеспечения) системы обмена сообщениями доступны далеко не на всех платформах и используют собственные закрытые протоколы. Примеры: *IBM Integration Bus, IBM MQ, Microsoft Message Queuing (или MSMQ), Microsoft Azure, Azure storage queues* и *AppFabric Service Bus, Oracle Messaging Cloud Service, OpenEdge Sonic MQ, TIBCO Enterprise Message Service*.

В тоже время развивались открытые проекты, каждый из которых имел своё целевое значение и решал задачи в достаточно узкой области. Со временем многие из них распространяли влияние на смежные области, повышался уровень абстракции предлагаемых решений. В результате часть из них приобрела популярность и значимость, как универсальные продукты для целых классов задач. В качестве иллюстрации вышесказанного в табл. 1 перечислим открытые проекты со свободно распространяемым исходным текстом программ, дадим их краткое описание [11].

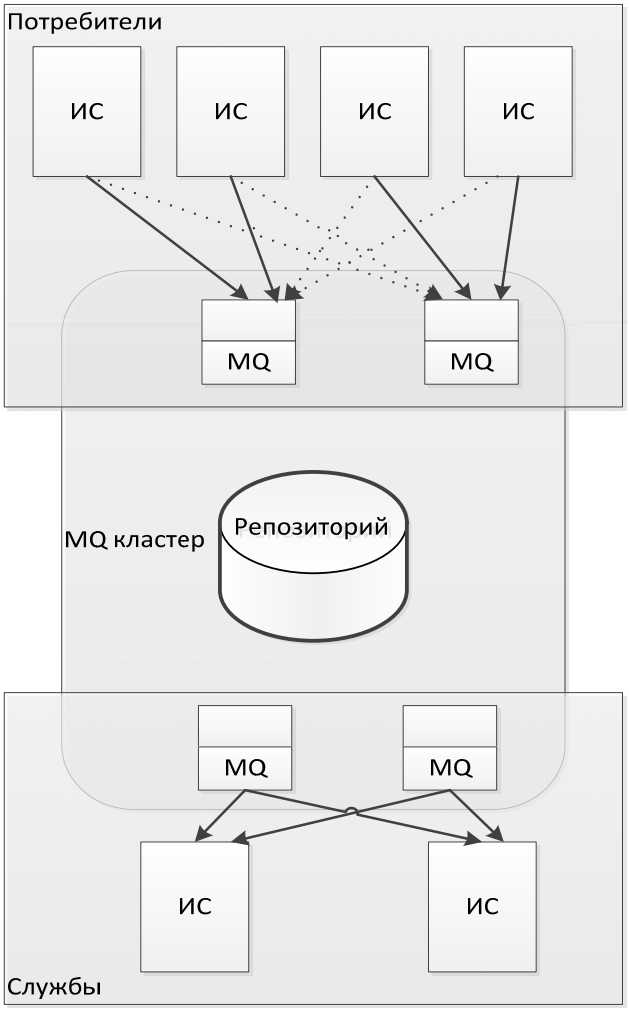


Рис. 2. Распределенная система с резервированием и использованием *MQ* кластера

Таблица 1

Системы с открытым кодом

№	Название	язык/платформа/ тип (services)	Описание
1	*ActiveMQ*	*java*	Самый популярный и мощный сервер обмена сообщениями и шаблонами интеграции с открытым исходным кодом [12]
2	*Amazon MQ*	*service*	Служба брокера управляемых сообщений для *Apache ActiveMQ*, которая упрощает настройку и работу брокеров сообщений в облаке [13]
3	*Amazon Simple Queue Service (SQS)*	*service*	Служба очереди сообщений, которая обрабатывает сообщения или рабочие потоки между другими компонентами в системе [14]

№	Название	язык/платформа/ тип (services)	Описание
4	*Apollo*	*java* *scala* *amqp*	Следующее поколение *ActiveMQ* [15]
5	*Beanstalkd*	*c*	Простая и быстрая в использовании очередь. Создана для улучшения времени отклика приложения «*Causes on Facebook*». Обеспечивает простой протокол, аналогичный *memcached*. Очереди имеют O (*log n*) операций *push* и *pop*. может быть запущен в постоянном режиме, который запишет все задания в *binlog*. Длинный список клиентских библиотек для многих языков [16]
6	*Beanstalkg*	*go*	*beanstalkd* на языке *Go* [17]
7	*Celery*	*python*	Распределенная система в основном для очереди задач и расписаний [18]
8	*Darner*	*cpp*	Простая, легкая очередь сообщений [19]
9	*Delayed::Job*	*ruby* *mysql*	Асинхронная система с поддержкой приоритетов [20]
10	*Disque*	*redis* *disque*	Распределенная система, работает на БД класса *in-memory redis* [21]
11	*Enduro/X Middleware platform*	*c* *cpp* *go*	Платформа распределённой обработки транзакций. Платформа предназначена для создания приложений на основе микросервисов в режиме реального времени с возможностью кластеризации. *Enduro / X* работает как расширенная замена *Oracle Tuxedo*. Платформа использует очереди ядра *POSIX* в памяти, что обеспечивает высокую пропускную способность межпроцессного взаимодействия [22]
12	*Faktory*	*go* *ruby* *rocksdb*	Высокопроизводительная обработка заданий [23]
13	*Gearman*	*c*	Универсальная прикладная среда для переноса нагрузки на другие машины или процессы [24]
14	*HornetQ*	*java* *amqp* *jms*	Проект с открытым исходным кодом для создания многопротокольной, встраиваемой, высокопроизводительной, кластерной, асинхронной системы обмена сообщениями (вошел в *Apache ActiveMQ*) [25]
15	*huey*	*python* *redis* *django*	Небольшая система для поддержки очереди заданий [26]
16	*IronMQ*	*Go* *service*	Простой в использовании высокодоступный сервис очереди сообщений. Доступен как облачный сервис на *Amazon* и *Rackspace*, а также локально с корпоративным предложением *Iron.io*. Функции включают в себя панель управления очередями, простые для создания *Webhook* (механизмов оповещения системы о событиях), одноадресные и многоадресные *push*-очереди, оповещения о необходимости масштабирования для рабочих процессов и очередей ошибок [27]
17	*Apache Kafka*	*scala*	Система обмена сообщениями «*publish-subscribe*» в виде распределенного журнала транзакций [28]
18	*Kue*	*node.js* *priority* *redis*	Распределенная очередь заданий с поддержкой приоритетов, поддерживаемая *redis*, созданная для *node.js* Характерные особенности: отложенные задания, дополнительные попытки соединения с отсрочкой, распределение параллельной рабочей нагрузки, поддержка режима *time-to-live* для задания, поддержка событий задания и сообщений типа «издатель/подписчик», богатый интегрированный интерфейс, использование *RESTful JSON API*, поддержка на *Redis* [29]
19	*Mappedbus*	*java* *high-throughput* *low-latency* *message-passing* *ipc*	Шина сообщений с высокой пропускной способностью и низкой задержкой на основе *Java*, использующая в качестве транспорта общий файл или область памяти [30]

№	Название	язык/платформа/ тип (services)	Описание
20	*Message Bus*	*hornetq* *java* *ruby*	Распределенная платформа обмена сообщениями, созданная на основе *HornetQ*, широко используемая в *Groupon* [31]
21	*nanomsg*	*c* *zeromq*	Библиотека сокетов, которая предоставляет несколько общих шаблонов связи [32]
22	*NATS*	*go* *.NET* *node* *nginx* *java* *ruby* *python* *scala*	Высокопроизводительная облачная система обмена сообщениями с открытым исходным кодом [33]
23	*NSQ*	*go*	Распределенная масштабируемая система обработки сообщений класса систем реального времени [34]
24	*QDB*	*java* *persistent* *replay* *backup*	Очередь сообщений с поддержкой воспроизведения [35]
25	*Apache Qpid*	*java* *amqp* *cpp* *Python, Java JMS* *.NET*	Инструменты для работы с системой обмена сообщениями с поддержкой протокола *AMQP*, поддерживают многие языки и платформы. *AMQP* – это открытый интернет-протокол для гарантированной отправки и получения сообщений [36]
26	*queue_classic*	*ruby* *postgres*	Простая, эффективная система управления очередей на *Ruby & PostgreSQL* [37]
27	*RabbitMQ*	*erlang* *amqp*	Надежная система обмена сообщениями для приложений, используется в *OpenStack* [38]
28	*Resque*	*ruby* *redis*	Очередь заданий, написанная на *Ruby* при поддержке *Redis*, работает на *GitHub*. Веб-интерфейс для управления запущенными заданиями, исполнителями и пр. Имеется поддержка плагинов (например, *Resque-Scheduler, Resque-Retry*). Широко используется [39]
29	*RestMQ*	*Python* *Cyclone* *redis*	Очередь сообщений, которая использует *HTTP* в качестве транспорта, *JSON* – в качестве протокола, работает как ресурсы *REST*, использует *Twisted* и *Redis* [40]
30	*RQ*	*python* *redis*	Библиотека *Python* для организации очередей заданий и их обработки в фоновом режиме [41]
31	*Siberite*	*go*	Реализация *Darner* на *Go* с дополнительными функциями. Простая и быстрая очередь сообщений с поддержкой *LevelDB*: указателей при многократном использовании процедуры, протокола *Kestrel* (*memcached*). Очередь сохраняет все сообщения процесса, характеризуется оптимальным объёмом занимаемой оперативной памяти независимо от размера очереди, поддержкой двухфазных запросов [42]
32	*Sidekiq*	*ruby* *crystal* *redis*	Простая, эффективная фоновая обработка сообщений для *Ruby*. Основано на использовании *Redis*. Характеризуется многопоточностью, использованием *Celluloid* (для *Ruby*), наличием *Web*-интерфейс, совместимость с *Resque*. *Sidekiq Pro* - платная версия с поддержкой и дополнительными функциями (пакеты, уведомления, надежность, метрики). Широко используется разработчиками [43]
33	*SnakeMQ*	*python*	Небольшая кроссплатформенная библиотека *Python* для простого и надежного взаимодействия между узлами сети [44]
34	*StormMQ*		Служба очереди сообщений, использует протокол расширенной очереди сообщений (*AMQP*) [45]

№	Название	язык/платформа/ тип (services)	Описание
35	*Zaqar (ex Marconi)*	*openstack python mongodb sqlite durable*	Служба очередей и уведомлений, созданная для *OpenStack*, используется не только в *OpenStack* [46]
36	*ZeroMQ*	*cpp java*	Распределённая система обмена сообщениями [47]

Подход DCPS

В подходе *DCPS* центром и средством взаимодействия являются данные, а не сообщения. В этом случае инфраструктура управляет данными, при помощи которых контролируется работа распределённых систем, определяет их структуру и формат, правила модификации и доступа. Используя подход, ориентированный на данные, разработчики пишут приложения, которые читают и обновляют записи в определённом таким образом пространстве, именуемом *Global Data Space* (*GDS*), или шина данных. Для корректной работы *DCPS* контролирует синтаксис данных в интеграционном слое (*middleware*). Составляющая интеграционного слоя, шина данных, ответственная за управление данными, может также состоять из набора специализированных приложений и систем:

- реализующих синтаксический, форматно-логический контроль;
- типа *Master Data Management* (MDM) [48] – совокупность процессов и инструментов для постоянного определения и управления основными данными (в том числе справочными), хранения и управления таксономиями и онтологиями объектов обмена;
- обеспечивающих работу с данными так же как и БД или хранилища данных, включая их загрузку, хранение, модификацию и доступ.

Аналогично хранилищу данных, корректно организованное пространство данных промежуточного слоя (*GDS*) [49] в силу своей центральной роли становится для распределённых систем источником достоверных данных и значительно облегчает системную интеграцию.

Приложения, которые хотят внести вклад в *GDS,* объявляют о своём намерении стать «издателем». Приложения, желающие получить доступ к частям этого пространства данных, объявляют о намерении стать «подписчиками». Каждый раз, когда издатель публикует новые данные в *GDS*, промежуточное программное обеспечение распространяет информацию всем заинтересованным подписчикам. В основе любой системы подписки на публикации, ориентированной на данные, лежит их модель. Она определяет *GDS* и указывает, как издатели и подписчики ссылаются на части этого пространства, что также позволяет промежуточному программному обеспечению обеспечить уровень безопасности использования данных. Целевые приложения часто требуют более высокого уровня модели данных, позволяющей выражать отношения агрегации и согласованности между элементами данных.

Характерные различия подходов MCPS и DCPS

В *DCPS* шина данных выполняет роль темпоральной базы данных (оперирующей с темпоральными данными, связанными с определенными датами или промежутками времени) для интегрируемых составных частей распределённой системы. В этом состоит отличие *DCPS* от *MCPS* [5] в случае обмена параметрами. Система *MCPS*, ориентированная на обмен сообщениями, например *MOM*, отправит *N* сообщений со значением параметра, получатель должен обработать каждое из них, отправив подтверждение. Промежуточное программное обеспечение, ориентированное на данные, *DCPS* распределяет пространство данных и предоставляет к ним доступ в соответствии с правилами. В обоих случаях приложения получают корректные значения, но сетевой трафик и работа приложений сильно различаются. В случае *DCPS*, промежуточное программное обеспечение обновляет свои структуры в соответствии с алгоритмами работы специализированных приложений и уведомляет подписчиков, зарегистрированных на получение обновлённых данных, о необходимости прочитать новое значение в соответствии с правилами подписки. Например, уведомляет подписчика только при выполнении зарегистрированных для него условий. Таким образом, инфраструктура сама контролирует и отвечает за состояние распределённой системы. Управление регистрируемых в промежуточном слое данных осуществляется параметрами, составляющими конфигурацию или установки «Качества обслуживания» (*QoS*) (см. рис.3)

Сравнивая подходы *MCPS* и *DCPS*, отметим, что они не являются альтернативой друг другу. Для реального внедрения может быть рассмотрен как каждый из них со своими сильными и слабыми сторонами, так и их объединение, которое также может быть хорошим выбором для сложных взаимосвязанных информационных систем цифровой экономики.

При проектировании интеграционного слоя, ориентированного на обмен сообщениями, в *MCPS* единицей обмена информацией является само сообщение. Роль инфраструктуры заключается в том, чтобы сообщения гарантированно доставлялись их предполагаемым получателям. При этом оно может содержать что угодно в любом формате. Инфраструктура просто передаёт информацию, а основное внимание уделяется обеспечению быстрых и надёжных бизнес-транзакций, в соответствии с подходом *ACID* отслеживанию всех сообщений и обеспечению гарантированной доставки каждого из них в соответствии с назначением, независимо от сбоев или перезагрузок.

Таблица 2

Различия *MCPS* и *DCPS*

MCPS	*DCPS*
Программное обеспечение инфраструктуры не контролирует и не разбирает данные	Программное обеспечение инфраструктуры контролирует и управляет сбором и хранением данных
Проверка синтаксиса, формирование и разбор сообщений происходит на уровне интегрируемых приложений, что повышает требования к конечным приложениям	Приложения только получают или отправляют данные в формате, согласованном с программным обеспечением инфраструктуры. Требования к конечным приложениям снижаются
Доставка сообщения происходит без разбора на уровне инфраструктуры	Постоянный контроль и управление данными в программном обеспечении инфраструктуры поглощает ресурсы промежуточного слоя

С другой стороны, практически любое распределённое приложение характеризуется множеством состояний интегрированных компонентов или автономных систем. Если инфраструктура не управляет состоянием системы, то каждое приложение должно хранить и поддерживать своё собственное состояние. При этом синхронизация состояний таких приложений выливается в отдельную задачу управления. Неуправляемое состояние быстро приводит к несогласованности и, как следствие, несовместимости систем, потере управления и функциональности, вплоть до полного разрушения данных. При подходе, ориентированном на данные – *DCPS*, управление состоянием приложений, компонентов и систем вынесено в отдельный слой программного обеспечения, при этом управление состоянием распределённой системы, а значит, и самой системы легче сделать надёжным, предсказуемым и доступным. В отличие от *MCPS*, при *DCPS*, интеграционный слой оправляет не сообщения о данных, а сами предварительно обработанные данные, работая с которыми, *DCPS* рассматривает их как экземпляры зарегистрированных в промежуточном слое объектов, оперируя, таким образом, данными, а не потоками информации.

Для реализации промежуточного программного обеспечения, ориентированного на подход *DCPS*, существует много стандартов и продуктов. Один из успешно зарекомендовавших себя – стандарт *Data Distribution Service* (*DDS*) [50], ориентированный исключительно на данные. В отличие от *MCPS* технологии *DCPS* в стандарте *DDS* поддерживают широкий набор гибко регулируемых параметров и установок качества обслуживания (*QoS)*. Наличие *QoS* обеспечивает надёжность и детерминированность информационного обмена. С помощью *QoS* источники и потребители «договариваются» между собой о наилучшем балансе интересов потребителя с возможностями источника по временным и иным характеристикам информационного потока.

Описанные подходы *MCPS* и *DCPS* могут применяться для интеграции систем. Однако подход *DCPS* понимает сами данные и поэтому может управлять различными типами поведения состояний. При этом подходе работа инфраструктуры заключается не только в доставке сообщений, но и в обеспечении синхронизированного и общего понимания всеми узлами значения данных. В этом работа инфраструктуры отчасти напоминает работу СУБД. Таким образом, реализация подхода *DCPS* обеспечивает лучшие по сравнению с *MCPS* масштабируемость и управление состоянием всей распределённой системы за счет переноса ряда критических функций в промежуточное программное обеспечение.

В табл. 2. представлены наиболее характерные различия подходов *MCPS* и *DCPS*.

СЛУЖБА РАСПРОСТРАНЕНИЯ ДАННЫХ

Спецификация службы распространения данных *Data Distribution Service* (*DDS*) [50] вводит стандарт на программный интерфейс прикладного программирования (*API*), с помощью которого распределённое приложение может использовать *DCPS* в качестве механизма связи.

DDS – это открытый стандарт для обмена сообщениями группы управления объектами, который поддерживает уникальные потребности как корпоративных систем, так и систем реального времени.

Ключевые особенности *DDS.*

- *Портативность*. *DDS* изначально разрабатывался для поддержки любого языка программирования, например, *C* и *C* ++; он также поддерживает интерфейсы *Java, C #, Ada, JMS, WSDL / SOAP* и *REST / http.*
- *Совместимость при обмене данными*. Протокол обмена *DDS Real-Time Publish-Subscribe* (*RTPS*) обеспечивает бесшовную совместимость между реализациями, платформами и языками программирования.
- *Семантическая совместимость*. Приложения обмениваются зарегистрированными объектами данных, которые обычно описываются стандартной моделью, указанной с использованием языка определения интерфейса *OMG* (*IDL*), *XSD* / *WSDL*, *XML* или программного интерфейса. Она также может быть сгенерирована из модели *UML.*

- *Множество реализаций*. По крайней мере 10 уникальных реализаций промежуточного программного обеспечения поддерживают *DDS API* или протокол обмена.

Кроме того, как указывалось ранее, *DDS* стандартизирует семантику обмена сообщениями, что снижает затраты на разработку и интеграцию, улучшая масштабируемость и надёжность системы.

Обмен, ориентированный на данные, позволяет задавать различные параметры, такие как скорость публикации, скорость подписки, срок действия и многие другие. Эти параметры качества обслуживания (*QoS*) позволяют разработчикам систем создавать распределенное приложение на основе требований и доступности каждого конкретного фрагмента данных. Таким образом, *DDS* определяет полный набор параметров качества обслуживания (*QoS*). Они обеспечивают управление динамическим обнаружением, маршрутизацией и фильтрацией с учетом содержания, отказоустойчивостью и детерминированным поведением в реальном времени. Среда, ориентированная на данные, позволяет иметь механизм связи, адаптированный к требованиям конкретного распределенного приложения.

Используя метод «публикация-подписка» для передачи данных, *DDS* обеспечивает абстрактную, опосредованную связь между отправителями и получателями. Издатели не обязаны знать о каждом отдельном получателе, они должны знать только о конкретном типе данных, которые передаются. То же самое относится и к подписчикам. Они не должны знать, откуда публикуются данные, им нужна только информация о конкретном типе получаемых данных.

Спецификации для *DDS* [51-55] разбиты на два отдельных раздела. Первый раздел посвящён публикации-подписке, ориентированной на данные (*DCPS*), а второй раздел – уровню локальной реконструкции данных *Data Local Reconstruction Layer* (*DLRL).*

DCPS – это программный интерфейс нижнего уровня, который может использовать приложение для связи с другими приложениями с поддержкой *DDS*. *DLRL* – это часть верхнего уровня спецификации, описывающая, как приложение может взаимодействовать с полями данных *DCPS* через свои собственные классы объектно-ориентированного программмирования. *DLRL* – необязательный уровень в спецификации *DDS*.

Интеграция действующих систем с заблаговременным предоставлением информации («публикацией») и своевременным ее отбором и потреблением («подпиской»), а именно, ее реализация в стандарте *DDS*, является и надёжной, и детерминированной, и масштабируемой. В отличие от других моделей взаимодействия, в которых в явном виде надо указывать, что кому и когда направить, модель *DDS* анонимная: источники не знают, какие узлы распределённой системы получают их информацию, а потребителям не известны, узлы, выступающие источниками информации, которую они получили. Эта анонимность является основой масштабируемости модели *DDS*: реконфигурация узлов не приводит к необходимости переписывать сетевые связи. Надёжность и детерминированность информационного обмена обеспечивается наличием в стандарте *DDS* набора параметров качества обслуживания (*QoS*), с помощью которых источники и потребители «договариваются» между собой о наилучшем балансе интересов потребителя и возможностями источника по временным и надежностным характеристикам информационного потока между парой «источник-потребитель».

DDS нашел широкое применение в высокопроизводительных оборонных, промышленных и встроенных в оборудование приложениях. Подход позволяет эффективно доставлять миллионы сообщений в секунду нескольким получателям одновременно. Кроме того, *DDS* обеспечивает управляемое соединение в режиме реального времени «многие ко многим», необходимое для высокопроизводительных приложений.

Спецификация *DDS* описывает модель публикации-подписки на основе данных (*DCPS*) для связи и интеграции распределённых приложений. Эта спецификация определяет как прикладные интерфейсы (*API*), так и семантику связи, поведение и качество обслуживания, обеспечивающие эффективную доставку информации от производителей потребителям.

Цель спецификации *DDS* может быть обобщена как «Эффективная и надёжная поставка корректной информации в нужное место в нужное время».

Области применения требуют, чтобы подход *DCPS* был высокопроизводительным и предсказуемым, а также эффективным в использовании.

Для удовлетворения требований к оптимальному использованию ресурсов важно, чтобы интерфейсы были спроектированы таким образом, чтобы они:

- позволяли промежуточному программному обеспечению предварительно распределять ресурсы, а динамическое распределение ресурсов могло быть сведено к минимуму;
- избегали ситуаций, когда могут потребоваться неограниченные или трудно предсказуемые ресурсы;
- сводили к минимуму необходимость копировать данные.

DDS использует типизированные интерфейсы, то есть интерфейсы, которые учитывают фактические типы данных в максимально возможной степени. Типизированные интерфейсы обладают преимуществами:

- *просты в использовании*. Программист напрямую манипулирует конструкциями, которые естественным образом представляют данные;
- *безопасны в использовании*. Проверки синтаксиса и формата выполняются во время компиляции;
- *эффективны*. Код выполнения опирается на знание точного типа данных, что позволяет предварительно распределить ресурсы.

Однако использование типизированных интерфейсов означает наличие средств для перевода описаний типов в соответствующие интерфейсы, а также программного обеспечения, заполняющего разрыв между типизированными интерфейсами и общим промежуточным программным обеспечением.

QoS (качество обслуживания) – это общая концепция *DDS*, используемая для определения работы службы. Программирование работы службы с помо-

щью настроек *QoS* приводит к тому, что разработчик приложения указывает только «что» нужно, а не «как» это сделать. *QoS* состоит из списка независимых политик *QoS*. Каждая политика *QoS* представляет описание, ассоциированное с парой: имя – значение.

Эта спецификация *DDS* предназначена для обеспечения четкого разделения между сторонами публикации и подписки, так что процесс приложения, в котором участвует только издатель, может включать только то, что относится к публикации. Точно так же процесс приложения, в котором участвует только подписчик, может встраивать только то, что относится к подписке [56,57].

Исторически для распределённых информационных систем был предложен подход, использующий посредника – брокера запросов от одной системы к другой. *Common Object Request Broker Architecture* (*CORBA)* – технологический стандарт написания распределённых приложений, продвигаемый консорциумом *OMG* (*Object Management Group* (*OMG*), который содержит требования к архитектуре брокера объектных запросов, или типовой архитектуре опосредованных запросов к объектам) [58,59],

По ряду объективных причин, связанных с трудностью реализации приложений по стандарту *CORBA*, разработчики *DDS* были вынуждены разработать и реализовать свои спецификации и реализацию приложений. В некоторых случаях *DDS* использует *CORBA* для администрирования и управления установлением подключений к данным, но никогда – при распространении данных и межпроцессорного взаимодействия.

Спецификация *DDS* определяет два отдельных интерфейса [59].

1. Описанный ранее интерфейс публикации-подписки, ориентированный на данные (*DCPS = Data-Centric Publish-Subscribe*) представлен на рис. 3. Он обеспечивает глобальное пространство данных, считается интерфейсом низкого уровня и очень похож на реализации *MOM*, которые были распространены в 1990-х годах.

2. Уровень локального восстановления данных *Data Local Reconstruction Layer* (*DLRL*). *DLRL* – дополнительный слой, построенный поверх *DCPS*, который реализует объектную модель данных, обеспечивает интерфейс более высокого уровня и скрывает большую часть деталей публикации/подписки за локальной структурой, обрабатываемой на уровне промежуточного слоя (рис. 4). Это возвращает *DDS* обратно к обеспечению большей прозрачности.

В обоих случаях видно, что явные преимущества в производительности достигаются, когда «издатели» и «подписчики» используют транспорт напрямую. Также очевидно, что *DLRL* предоставляет платформу, которая проще для использования непосредственно приложениями. Как и в случае *DDS DCPS*, *DDS DLRL* использует типизированные интерфейсы.

В проекте *OpenDDS* [60] в настоящее время реализуется уровень *DCPS*.

DCPS определяет функциональность, используемую приложением для публикации и подписки на значения объектов данных, что позволяет:

- регистрировать приложения для идентификации объектов, которые они намерены публиковать, а затем предоставлять данные для этих объектов;
- регистрировать приложения и объекты, которые им «интересны», а затем предоставлять доступ к данным этих объектов;
- определять список тем, при помощи которых помечается информация, политики *QoS*, сущности.

Спецификация *DDS* определяет две модели: платформенно-независимую модель (*PIM – Platform Independent Model*) и платформу (*PSM – Platform Specific*), построенную на основе PIM, но ориентированную на использование спецификаций на *OMG IDL* (*CORBA*).

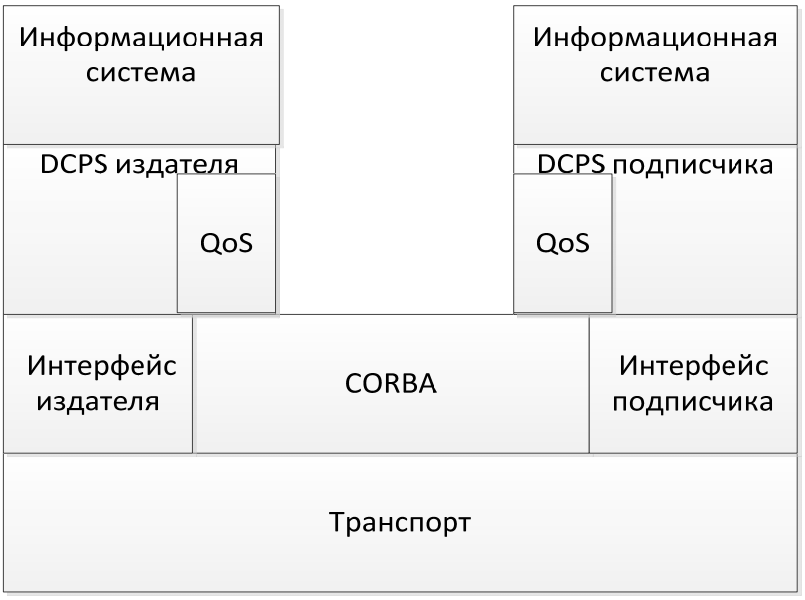


Рис. 3. Интерфейс публикации подписки, ориентированный на *DCPS*

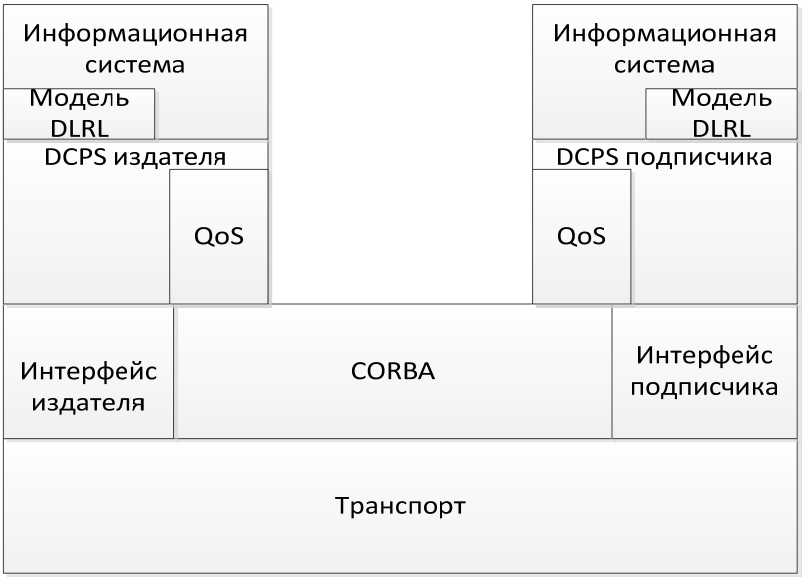


Рис. 4. Интерфейс публикации подписки, ориентированный на *DLRL*

Спецификация *DDS* предлагает программный интерфейс, позволяющий приложениям получать доступ к данным в логическом глобальном пространстве *Global Data Space* (*GDS*), или шине данных, о которой речь шла ранее. Интерфейсы *API DDS* обеспечивают прямой доступ к этим «общим данным» единообразным образом, что означает, что все приложения имеют общее представление о них с точки зрения адресации в темах и схемах. Однако некоторым приложениям требуется локальное представление этих данных, которое организовано так, чтобы соответствовать цели и бизнес-логике приложения. Таким образом, каждый процесс, который обращается к глобальным данным., отличается от другого. Спецификация *DLRL* решает эту проблему, предоставляя удобный локально определенный программный интерфейс, который абстрагирует доступ к распределенной информации. Уровень локального восстановления *DLRL* автоматически восстанавливает данные из обновлений, предоставляемых *DDS*, позволяя приложению получать к ним доступ так, как если бы они были локальными. Следовательно, комбинация *DLRL* в *DDS* не только распространяет информацию всем заинтересованным подписчикам, но также обновляет локальную копию информации в локальном формате, указанном каждым приложением.

В рамках модели *DLRL* разработчик приложения может:

- описать классы объектов с их методами, полями данных и отношениями;
- присоединить некоторые из этих полей данных к объектам *DCPS*;
- управлять этими объектами (создавать, читать, писать, удалять);
- гарантировать при управлении объектами неизменность ссылок.

Спецификация *DLRL* определяет:

- какие объектно-ориентированные конструкции можно использовать для определения объектов *DLRL*;
- какие функции применимы к этим объектам (например, создать, удалить и т.д.);
- различные уровни отображения между двумя слоями:
  - структурное отображение, отношения между объектами *DLRL* и данными DCPS;
  - операционное сопоставление, т. е. сопоставление объектов *DLRL* с объектами *DCPS* (*Publisher, DataWriter* и т.д., включая настройки *QoS*, комбинированные подписки;
  - функциональное отображение, т. е. отношения между функциями DLRL (в основном, доступом к объектам *DLRL*, и функциями *DCPS*, запись / публикация.

DLRL позволяет на уровне приложения описать: методы; атрибуты, которые могут быть:

- локальными, не участвующими в распространении данных;
- общими, участвующими в процессе распространения данных и, следовательно, привязанными к объектам *DCPS*.

КРИТЕРИИ ВЫБОРА ТЕХНОЛОГИЙ

У всякого технологического решения есть два пути: либо решение становится всё более востребованным, популярным и развивается, либо его участь ограничивается сравнительно небольшим сроком эксплуатации автоматизированной системы, для которой оно было разработано – обычно это не более десятка лет.

Для технологических решений цифровой экономики крайне желателен первый путь, как наименее затратный. Проследим на примере Интернета [9], какими причинами может быть обусловлено успешное продвижение технологии. Перечислим некоторые из них.

1. Сравнительная простота, а, значит, и технологическая доступность заложенной конструкции. Участники Интернета сами выступают и в качестве потребителей, и в качестве разработчиков, и в качестве поставщиков. Интернет помимо средства информационного обмена стал и общедоступной платформой производства.

2. Вследствие простоты конструкции ее развитие не требует пересмотра базовых технологий и должно допускать безусловную совместимость различных версий на протяжении длительного периода развития.

3. Отсутствие оплаты. Все основные стандарты, протоколы, технологии абсолютно открыты, бесплатны и развиваются при участии огромного количества людей.

4. Зрелость технологий, применяемых в качестве основы. Интернет использовал проверенные, в том числе временем, технологические решения.

5. Наличие проверенных, отлаженных в технологическом смысле стандартов.

6. Децентрализация и распределенность. Интернет – элемент инфраструктуры современной технологической стороны жизни общества. Как элемент инфраструктуры, он не может и не должен быть централизованным, как любая традиционная транспортная сеть, например, дороги.

7. Масштабируемость.

Таким образом, наиболее выгодным решением для цифровой экономики, как зарождающейся инфраструктуры, будет то, которое бесплатно для использования, востребовано, просто, надёжно, доступно, открыто, отлажено, масштабируется, развивается по стандартам, которые обеспечивают преемственность по отношению к развитию.

ЗАКЛЮЧЕНИЕ

Для решения задачи управления распределённой инфраструктурой взаимодействующих систем целесообразно применять оба подхода: *MCPS* и *DCPS*. В рамках первого реализуется выполнение требований *ACID*, транзакций, в рамках второго – управление инфраструктурой информационных систем, как единым целым.

При выборе программной реализации подходов необходимо руководствоваться спецификой области автоматизации и сложившимися традициями, требованиями к инфраструктуре, наличием готовых реализаций, требованиями стандартов. На данный момент сложно выделить бесспорного лидера в рассмотренных подходах. Однако наблюдение за историей развития программного обеспечения для построения информационных инфраструктур позволяет сделать вывод, что привлекательными оказываются наиболее простые и надёжные открытые технологии, выполненные в соответствии с требованиями имеющихся стандартов.

СПИСОК ЛИТЕРАТУРЫ

1. Грушо А.А., Забежайло М.И., Зацаринный А.А., Писковский В.О. Безопасная автоматическая реконфигурация облачных вычислительных сред // Системы и средства информатики. – 2016. – Т. 26, № 3. – С. 83-92.
2. Грушо А.А., Забежайло М.И., Зацаринный А.А., Николаев А.В., Писковский В.О., Сенчило В.В., Судариков И.В., Тимонина Е.Е. Об анализе ошибочных состояний в распределенных вычислительных системах // Системы и средства информатики. – 2018. – Т. 28, № 1. – С. 101-111.
3. Грушо А. А., Забежайло М.И., Зацаринный А.А., Николаев А.В., Писковский В.О., Тимонина Е. Е. Классификация ошибочных состояний в распределенных вычислительных системах и источники их возникновения // Системы и средства информатики. – 2017. – Т. 27, № 2. – С. 30-41.
4. Hohpe G., Woolf B. Enterprise integration patterns: designing, building, and deploying messaging solutions. – Boston, MA: Addison-Wesley Longman Publishing Co., Inc., 2003.
5. Curry E. Message-Oriented Middleware. In: Middleware for Communications / Mahmoud Qusay H. (Ed.). – Chichester: John Wiley & Sons, 2004. – P. 1-28.
6. Chapter 4: Mocking and simulating JMS IBM® WebSphere MQ. – URL: https://trafficparrot.com/tutorials/mocking-or-simulating-jms-ibm-websphere-mq.html
7. Keen M., Acharya A., Bishop S., Hopkins A., Milinski S., Nott C., Robinson R., Adams J., Verschueren P. Patterns: Implementing an SOA Using an Enterprise Service Bus. – Redbook, 2004. – URL: https://www.redbooks.ibm.com/redbooks/ pdfs/sg246346.pdf.
8. IBM MQ. – URL: https://developer.ibm.com/ messaging/ibm-mq/
9. Тенденции и принципы проектирования сложных систем. Учебное пособие. – URL: https://mydocx.ru/7-53737.html
10. Kuhn R., Dyksy Z., Karlsson M. Deliverable D8.3 - Description of Integration Layer and Constituents, 2018. – URL: http://www.in2rail.eu/Page.aspx?CAT=DELIVERABLES& Id-Page=69d2e365-3355-45d4-bb3c-5d4ba797a3ac
11. Queues. – URL: http://queues.io
12. Apache ActiveMQ. – URL: http://activemq.apache.org/
13. Amazon MQ. – URL: http://aws.amazon.com/amazon-mq/
14. Amazon Simple Queue Service. – URL: http://aws.amazon.com/sqs/
15. ActiveMQ Apollo. – URL: http://activemq.apache.org/apollo/
16. Beanstalk. – URL: http://kr.github.io/beanstalkd/
17. Beanstalk. – URL: https://github.com/vimukthi-git/beanstalkg/pulls
18. Celery: Distributed Task Queue. URL: http://www.celeryproject.org/
19. Darner. – URL: https://github.com/wavii/darner
20. Delayed_job. – URL: https://github.com/collectiveidea/delayed_job
21. Disque. – URL: https://github.com/antirez/disque
22. Enduro/X. – URL: http://www.endurox.org/
23. Faktory. – URL: https://github.com/contribsys/faktory
24. Gearman. – URL: http://gearman.org
25. HornetQ. – URL: http://www.jboss.org/hornetq
26. Huey. – URL: https://huey.readthedocs.org/en/latest/
27. MQ. – URL: http://www.iron.io/mq
28. Kafka. – URL: http://kafka.apache.org/

29. Kue. – URL: https://github.com/Automattic/kue
30. Mappedbus. – URL: http://mappedbus.io/
31. Message-Bus. – URL: https://github.com/groupon/Message-Bus
32. Nanomsg. – URL: http://nanomsg.org/
33. NATS. – URL: https://nats.io
34. Nsq. – URL: https://github.com/bitly/nsq
35. QDB. – URL: http://qdb.io/
36. Apache Qpid. – URL: http://qpid.apache.org/
37. Queue_classic. – URL: https://github.com/ryandotsmith/queue_classic
38. RabbitMQ. – URL: http://www.rabbitmq.com/
39. Resque. – URL: https://github.com/resque/resque
40. RestMQ.– URL: http://restmq.com/
41. RQ. – URL: http://python-rq.org/
42. Siberite. – URL: http://siberite.org/
43. Sidekiq. – URL: http://sidekiq.org/
44. SnakeMQ. – URL: https://pypi.org/project/snakeMQ/
45. StormMQ. – URL: https://github.com/stormmq
46. Zaqar. – URL: https://wiki.openstack.org/wiki/Zaqar
47. ZeroMQ. – URL: http://www.zeromq.org/
48. MDM – Master Data Management – Управление основными мастер-данными. – URL: http://www.tadviser.ru/index.php/MDM_-_Master_Data_Management
49. Corsaro A., Schmidt D.C. The Data distribution service. the communication middleware fabric for scalable and extensible systems-of-systems. – URL: https://www.dre.vanderbilt.edu/~schmidt/PDF/dds-sos.pdf
50. AL-Madania B., Alib H. Data Distribution Service (DDS) based implementation of Smart grid devices using ANSI C12.19 standard // Procedia Computer Science. – 2017. – Vol. 110. – P. 394–401.
51. Data Distribution Service for real-time systems. Version 1.2. / OMG Available Specification formal/07-01-01. – 2007. – URL: https://www.omg.org/spec/DDS/1.2/About-DDS/
52. The Real-time Publish-Subscribe Protocol (RTPS) DDS Interoperability Wire Protocol Specification. Version 2.2. /OMG Document Number: formal/2014-09-01, Standard document. – 2014. – URL: http://www.omg.org/spec/DDSI-RTPS/2.2
53. Data Distribution Service (DDS). Version 1.4 / OMG Document Number: formal/2015-04-10. – Standard document. – 2015. – URL: http://www.omg.org/spec/DDS/1.4
54. Data Distribution Service (DDS), Version 1.4 with change bars / OMG Document Number: formal/2015-04-11. – Standard document. – 2015. – URL: http://www.omg.org/spec/ DDS/1.4
55. DDS Data Local Reconstruction Layer (DDS-DLRL). Version 1.4 / OMG Document Number: formal/2015-04-12, Standard document. – 2015. – URL: http://www.omg.org/spec/DDS-DLRL/1.4
56. Alaerjan A., Kim D.K. Configuring DDS features for communicating components in smart grids // 5th IEEE SEGE. – 2017. – URL: https://www.researchgate.Net/publication/319357595_Configuring_DDS_Features_for_Communicating_Components_in_Smart_Grids/download
57. Ваньков А.И. Технология DDS для создания систем передачи данных в распределенной электроэнергетике и сетях «Smart Grid». – М.: РусБИТех, 2017. – URL: http://digitalsubsta-tion.com/wp-content/uploads/2018/07/Vankov-A.I.-Tehnologiya-DDS-dlya-sozdaniya-sistem-peredachi-dannyh-v-raspredelennoj-elektroenergetike-i-setyah-Smart-Grid.pdf
58. Демьянов А.В. Связующее ПО стандарта DDS на земле, на воде и в воздухе. – URL: http://bp-la.ru/svyazuyushhee-po-dds/
59. OMG CORBA. – URL: http://www.corba.org/
60. OpenDDS. – URL: http://opendds.org/

Материал поступил в редакцию 25.03.19.

Сведения об авторах

ГРУШО Александр Александрович – доктор физико-математических наук, профессор, заведующий лабораторией ИПИ, ФИЦ ИУ РАН
e-mail: grusho@yandex.ru

ЗАБЕЖАЙЛО Михаил Иванович – доктор физико-математических наук, доцент, заведующий лабораторией ИПИ, ФИЦ ИУ РАН
e-mail: m.zabezhailo@yandex.ru

ПИСКОВСКИЙ Виктор Олегович – кандидат физико-математических наук, старший научный сотрудник ИПИ, ФИЦ ИУ РАН
e-mail: vpvp80@yandex.ru

СЕНЧИЛО Владимир Викторович – научный сотрудник научный сотрудник ИПИ, ФИЦ ИУ РАН
e-mail: volodias@mail.ru

ТИМОНИНА Елена Евгеньевна – доктор технических наук, профессор, ведущий научный сотрудник ИПИ, ФИЦ ИУ РАН
e-mail: eltimon@yandex.ru

АВТОМАТИЗАЦИЯ ОБРАБОТКИ ТЕКСТА

УДК 81’322

Ф.В. Краснов, М.Е. Шварцман, А.В. Дементов, А.И. Сень

Тематическая когерентность двуязычного корпуса научных статей (на примере нефтегазовой отрасли)

Изучены структурные различия научных статей при переводе с русского языка на английский. Использована методика модального тематического моделирования. В собранной коллекции каждый документ представлен двумя модами. В результате построения тематической модели были получены бимодальные матрицы Φ и Θ. Анализ матрицы Φ показал, что тематики разделились по степени соответствия между русскими и английскими терминами при рассмотрении слов в порядке убывания вероятности. Для 90% тематик английские слова полностью соответствовали русским. Анализ матрицы Θ показал, что для 99% документов существует тематика со значением больше 0,95. Таким образом, большинство документов являются монотематичными, что не зависит от языка документа.

Ключевые слова: *тематическое моделирование, последовательная регуляризация, ARTM, кластеризация текстов*

ВВЕДЕНИЕ

Официальным языком научных статей является английский, но достаточно большой объем научных работ изначально публикуется на родном языке учёного и только потом переводится на английский в более полном и углублённом виде. Следовательно, можно говорить о двуязычном корпусе документов.

В настоящее время в компьютерной лингвистике растёт интерес к двуязычным корпусам текстов для создания моделей машинного перевода. Например, в [1] применяется *Proceedings of the Canadian Parliament* на английском и французском, а в [2, 3] используются субтитры из кинофильмов на нескольких языках. Среди особенностей таких параллельных корпусов в [4] выделяют принадлежность языка определённой области и неполное соответствие смыслов переводов – особенности перевода. Анализу специфики параллельного перевода посвящено исследование [5], в котором отмечают несколько уровней параллельности: на уровне слов, фраз, предложений и рассуждений. В работах [1–4] авторы уделяют внимание как выделению пар совпадающих предложений, так и выстраиванию соответствий между словами. Такой подход является важным этапом в решении задачи статистического машинного перевода (СМП), которая была сформулирована более 50 лет назад в [6]. Важными вехами в истории создания подходов, основанных на СМП, являются создание моделей I и II центром исследований *IBM Watson* в 70-х гг. прошлого века[1].

В настоящее время существенным выглядит прогресс, достигнутый в [7, 8] с помощью архитектуры *Encoder-Decoder,* концепция которой подразумевает, что одновременное обучение производится для двух искусственных нейронных сетей, соединенных в виде воронки (выход одной, является входом для другой), а в центре находится скрытое представление переводимого текста *v*. Схема архитектуры *Encoder-Decoder*, представленная на рис. 1, адаптирована авторами настоящей статьи по материалам исследования [9].

Авторы настоящего исследования поставили цель – рассмотреть структуру скрытого представления *v*, применив в качестве *Encoder* и *Decoder* не искусственные нейронные сети, а аппарат тематического моделирования текста с последовательной регуляризацией [10].

Перевод научной статьи с русского на английский может быть выполнен по-разному. Некоторые авторы используют для перевода средства СМП, а некоторые пишут статью заново на английском. Результаты таких подходов отличаются. В случае использования СМП иногда говорят, что переводная статья написана на «русском английском» языке. Настолько это заметно для человека. На наш взгляд интерес представляет определение способов перевода статей.

[1] https://en.wikipedia.org/wiki/IBM_alignment_models

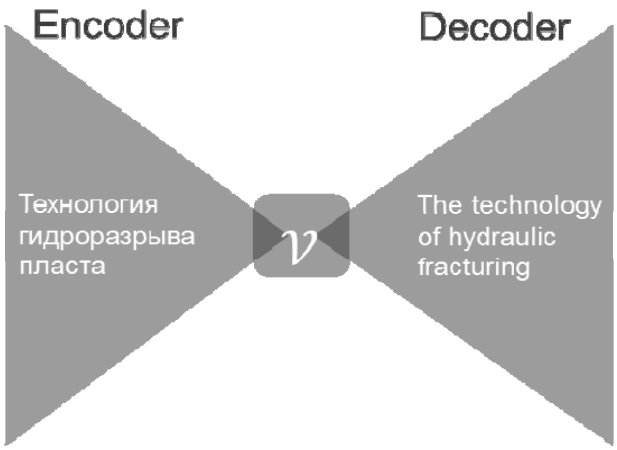


Рис. 1. Модель *Encoder-Decoder* для задачи статистического машинного перевода.

Сформулируем исследовательскую гипотезу нашей работы: для двуязычного корпуса научных статей возможно автоматическое выделение пары статей (одна – на языке автора, вторая – ее перевод) в которой перевод сделан при помощи средств СМП.

МЕТОДИКА ИССЛЕДОВАНИЯ

Для формальной оценки качества нескольких машинных переводов используют метрику *BLEU* [11]. Но в нашем случае она не подходит, так как мы имеем только один вариант перевода. Можно также рассмотреть гипотетический способ для проверки выдвинутой исследовательской гипотезы, при котором носителем английского языка выполняется перевод каждой статьи, а потом производится сравнение его с машинным вариантом. Но такой способ требует значительного количества ресурсов. И мы не сможем оценить варианты перевода, при которых авторы расширили свою научную статью. С точки зрения метрики *BLEU* они не являются точными.

С другой стороны, можно наоборот выполнить перевод русскоязычных научных статей на английский с помощью таких средств СМП, как *Moses* [12] или *Phrasal* [13], сделав при этом *baseline*-оценку, отклонения от которой допустимо измерять с помощью метрики *BLEU*. Но для того, чтобы обучить СМП, нужен большой двуязычный корпус научных статей по определённой тематике, которого у авторов статьи нет.

Для небольшого набора данных была выбрана предложенная в [10] методика выделения тематик из текста с помощью последовательной регуляризации, предусматривающая сначала обучение модели на каждом одноязычном корпусе документов отдельно, а затем сравнение выделенных тематик для каждой пары документов.

Суть тематического моделирования состоит в разложении векторного представления текста с помощью двух матриц (1):

$$p(w|d) = \sum_{t \in T} \varphi_{wt} \theta_{td} \qquad (1),$$

где φ_{wt} – матрица вероятностей слов w в каждой тематике $t \in T$; θ_{td} – распределение вероятностей тематик t в документе d; $p(w|d)$ – условная вероятность слова w в документе d.

Для обучения этой модели использовался механизм минимизации кросс энтропии с последовательным добавлением регуляризирующих слагаемых. Без регуляризации матрицы φ_{wt} и θ_{td} не представляют практического интереса.

Во множество тематик T целесообразно включить тематики двух типов – основные (sbj_i) и шумовые (nz_j). К шумовым можно отнести введение и обзор научных источников. Например, большинство статей по статистическому машинному переводу будут цитировать во введении одни и те же фундаментальные для этой области знаний научные работы, хотя основная тема статей будет отличаться. Для шумовых тематик нами проведена регуляризация со сглаживанием, для основных – регуляризация с разрежением. Таким образом, в последних снижен уровень шума.

Подбор регуляризационных коэффициентов τ был выполнен по методике, описанной в [14]. После обучения с регуляризацией матрицы φ_{wt} и θ_{td} стали разряженными на 80%.

Содержание матриц φ_{wt}^{rus} и φ_{wt}^{eng} представляет распределение тематик для каждого документа. Таким образом, для анализа скрытого представления переводимого текста v достаточно проанализировать соответствие тематик для разных языков. При этом возникает задача перевода одного названия темы с русского на английский. Ввиду высокой степени разрежённости матрицы φ объем этих работ будет небольшим и может быть выполнен с помощью электронного словаря по нефтегазовой тематике.

При сопоставлении φ_{wt}^{rus} и φ_{wt}^{eng} возникает задача выравнивания тематик для v. Другими словами, одна тематика на русском может соответствовать одной или нескольким на английском или оставаться без соответствия. Сложность такого выравнивания в матричном представлении описывается квадратной матрицей с размерностью $(dim\ T)^2$.

На основании полученного матричного представления v можно провести классификацию связей между статьями на разных языках.

ЭКСПЕРИМЕНТ

Особенностью настоящего исследования является небольшое количество документов – 484 документа (242 на русском и 242 на английском) загруженных с портала *OnePetro.org* международного сообщества нефтегазовых инженеров (*SPE*). Соответствие статей на разных языках установлено по индексу *DOI*. Выбор такого корпуса документов обусловлен тематической сфокусированностью. Все статьи согласно рубрикатору сосредоточены на одном научном направлении – гидродинамический разрыв пласта. Таким образом, для проверки предложенной гипотезы авторами был сформирован двуязычный корпус документов.

При создании словарей была применена лемматизация и отброшены высоко- и низкочастотные слова. Размер словарей для русского и английского корпуса подобран одинаковый – около $0.5 * 10^5$ слов.

Обучение модели останавливалось при достижении пологого характера изменений метрики перплексии (*Perplexity)*:

$$P = \exp\left(-\frac{1}{n}\sum_{d\in D}\sum_{w\in W} n_{dw} \ln\left[\sum_{t\in T}\varphi_{wt}\theta_{td}\right]\right), \qquad (2)$$

характеризующей информационную энтропию модели.

На рис. 2 показано, что энтропия русского текста выше энтропии английского. Сравнение значений *Perplexity* для разных языков находятся в согласии с результатами, опубликованными в работе [15]. Метрика *Perplexity* достаточно сильно зависит от редких слов в словаре.

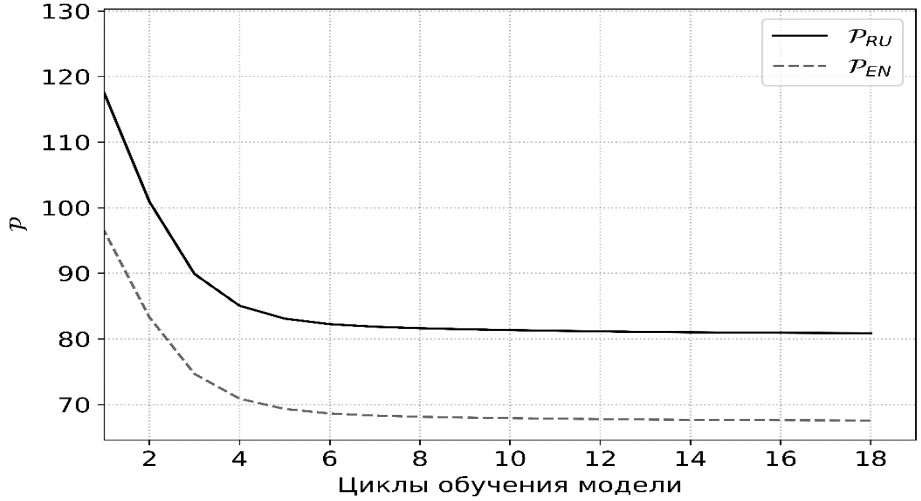


Рис. 2. Зависимость метрики *Perplexity* для русскоязычных и англоязычных статей от циклов обучения модели

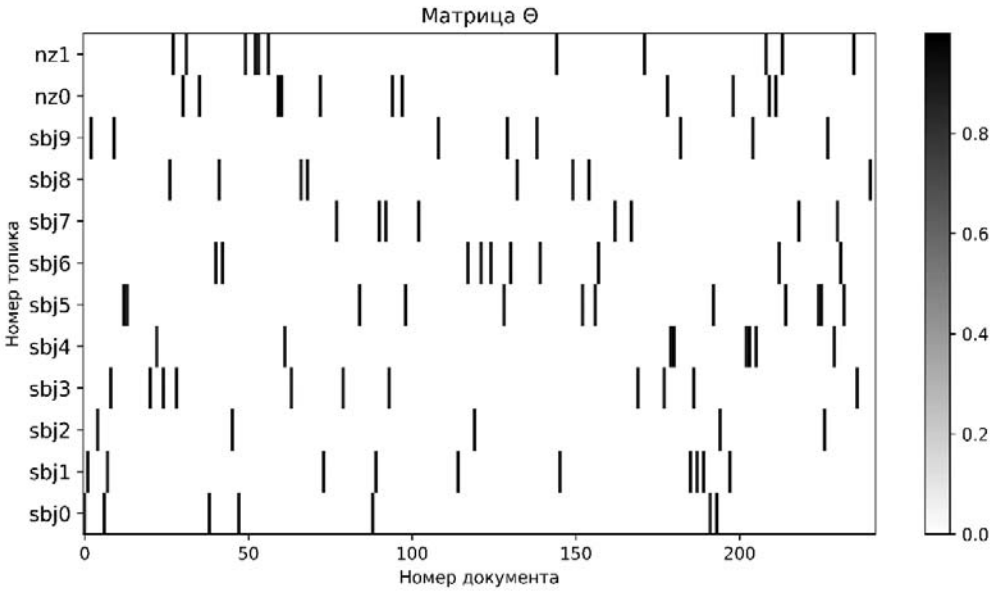


Рис. 3. Матрица значений θ_{td} после обучения модели.

Таблица с тематиками из матриц φ_{wt}^{rus} и φ_{wt}^{eng}

sbj0:RU	реакция, колонна, приток, насос, раствор, обсадный, концентрация
sbj0:EN	reaction, casing, pump, logging, log, mixture, string
sbj1:RU	долото, управление, дкс, наземный, интегрировать, мощность, пара
sbj1:EN	bit, steam, network, integrated, bcs, run, risk
sbj2:RU	трещина, грп, ячейка, приток, буровой, сетка, раствор
sbj2:EN	fracture, grid, equation, horizontal well, hydraulic, liner, boundary
sbj3:RU	трещина, напряжение, грп, модуль, геомеханический, упругий, трещиноватость
sbj3:EN	fracture, stress, medium, hydraulic, elastic, geomechanical, fracturing
sbj4:RU	приток, колонна, установка, оборудование, заканчивание, песок, труба
sbj4:EN	sand, esp, completion, inflow, pump, failure, tubing
sbj5:RU	кислотный, воздействие, отложение, кислота, залежь, разрез, карбонатный
sbj5:EN	acid, treatment, stimulation, carbonate, acid treatment, pilot, seismic
sbj6:RU	грп, трещина, микросейсмический, мгрп, проппант, порт, гнкт
sbj6:EN	fracturing, fracture, hydraulic, hydraulic fracturing, proppant, microseismic, port
sbj7:RU	потеря, интегрировать, сбор, расход, управление, нагнетательный, ппд
sbj7:EN	pipeline, integrated, flow rate, condensate, unit, gathering, reservoir pressure
sbj8:RU	раствор, буровой, колонна, буровой раствор, геомеханический, строительство, риск
sbj8:EN	mud, casing, geomechanical, risk, weight, history, stress
sbj9:RU	неопределенность, оторочка, рабочий, залежь, адаптация, конденсат, вытеснение
sbj9:EN	uncertainty, composition, history, condensate, mineral, sample, assessment
nz0:RU	образец, керн, эксперимент, раствор, частица, фильтрация, заводнение
nz0:EN	sample, core, cement, experiment, filtration, pore, strength
nz1:RU	образец, керн, карбонатный, пора, смачиваемость, гис, поровый
nz1:EN	pore, core, sample, carbonate, logging, wettability, space

Примечание: дкс – марка алмазного долота; грп – гидравлический разрыв пласта, мгрп – многостадийный гидравлический разрыв пласта; гнкт – гибкие насосно-компрессорные трубы; ппд – поддержание пластового давления; гис – геофизические исследования скважин

Для регуляризации были использованы коэффициенты μ, подобранные в работе [14]. После двенадцати итераций обучения с последовательной регуляризацией получена матрица θ_{td}, представленная на рис. 3, где по оси y тематики: sbj_{0-9} – основные и nz_{0-1} – шумовые. За счет разнонаправленной регуляризации пространство основных тематик разрежено, а пространство шумовых сглажено.

В таблице представлены примеры соответствий тематик для русского и английского корпусов. Из таблицы видно, что тематики на английском и русском для многих t_i полностью совпадают. Этот результат визуального анализа говорит о том, что тематическая модель настроена на существующую зависимость в данных. Но еще есть расхождения, которые надо анализировать.

Значения θ_{td} рассмотрим на рис. 4. Плотность максимальных значений θ_{td} для каждого документа отображена на рис. 4*a,* где показано, что существуют два характерных класса документов, которые можно разделить по значению максимальной θ_{td}. Изучим подробнее документы, с диаметрально противоположными значениями максимальной θ_{td}: 0,4101 – документ №214 и 0,9998 – документ №53. Из рис. 4*b* видно, что документ №53 укладывается в одну тематику sbj6, а документ №214 распределён по трём тематикам: sbj2, nz0 и nz1.

Сравним веса термов каждой из этих тематик, чтобы понять насколько они коррелированы. На рис. 5 отображены десять наибольших значений из матрицы φ_{wt} для тематик sbj6, sbj2, nz0 и nz1. Вероятности термов для тематики sbj6 значительно выше, чем для sbj2, nz0 и nz1. Это наблюдение свидетельствует о различном характере переводов документов.

Таким образом, исследуемая коллекция содержит документы двух разных типов, которые можно разделить по характеру распределения в них тематик. Большинство составляют документы, содержащие одно значение θ_{td} большее 0,95.

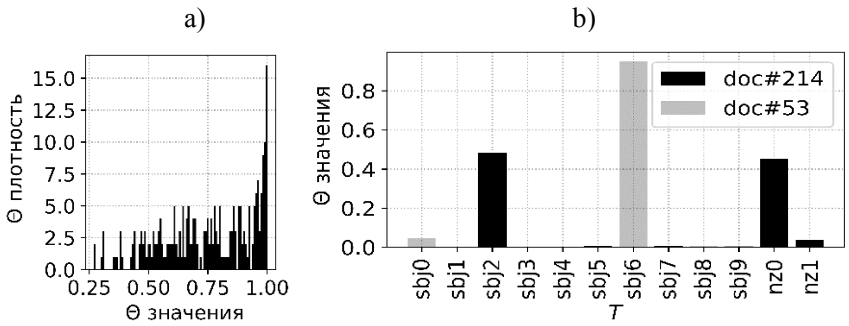


Рис. 4. Значения θ_{td} : *a* – плотность значений после обучения модели; *b* – значения для разнотипных документов.

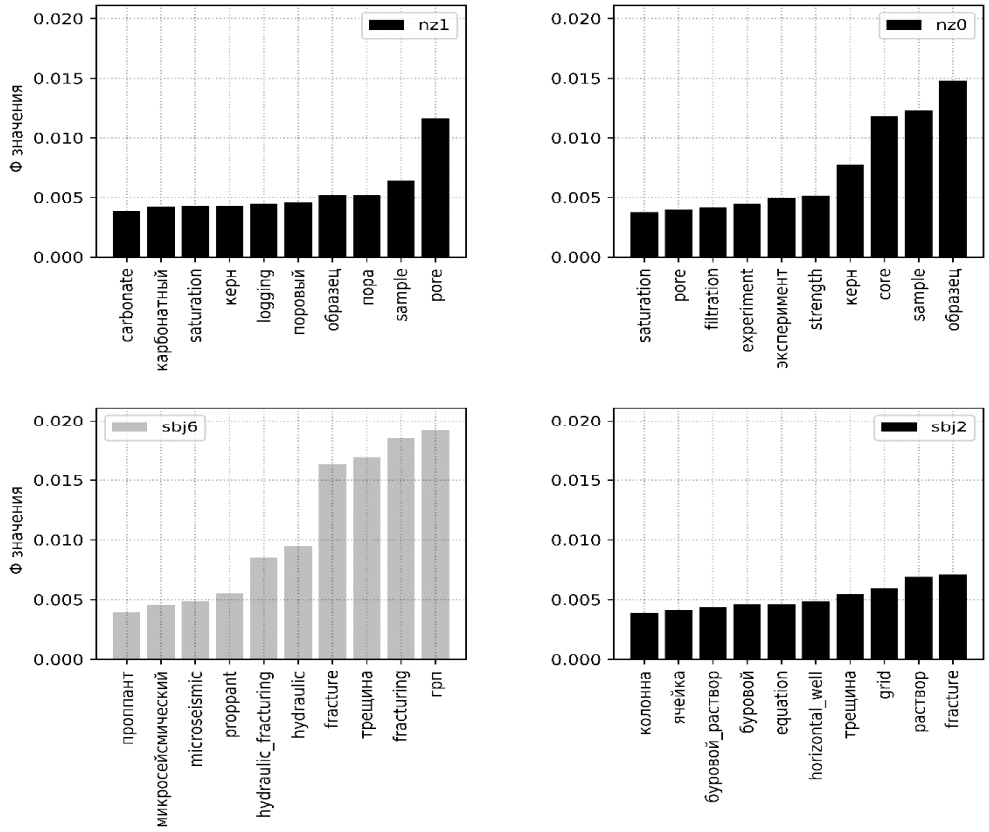


Рис. 5. Значения φ_{wt} для двух разнотипных документов.

ЗАКЛЮЧЕНИЕ

Авторами проведено исследование двуязычного корпуса документов. Наша основная цель заключалась в выделении особенностей перевода статей. Необходимость создания двух версий статьи (на русском, а после на английском) приводит к тому, что перевод может быть выполнен с помощью различных средств (профессиональный перевод, статистический машинный перевод). Наша гипотеза заключалась в том, что с помощью тематического моделирования с последовательной регуляризацией возможно выделить применные к переводу подходы.

В случае использования авторами для перевода средств СМП соответствие между английскими и русскими словами в тематиках должно быть более точным, чем при профессиональном переводе.

Гипотеза была подтверждена проведенным экспериментом. Переводная статья, написанная на «русском английском» языке, т.е. с помощью средств СМП, может быть точно идентифицирована на основе тематической модели.

В исследовании обнаружена кластеризация тематик аддитивной мультимодальной модели для корпуса двуязычных документов. Документы с типом перевода «русский английский» можно отличить от

документов с более творческим подходом к переводу по значениям φ_{wt} и θ_{td}.

Предложенная в статье методика подтверждает возможности использования стратегий регуляризации тематических моделей для получения компактных представлений тематик, подчеркивающих определенные характеристики исследуемой коллекции документов. Полученный результат может быть использован для автоматизированного определения качества перевода научных текстов.

СПИСОК ЛИТЕРАТУРЫ

1. Brown P.F., Pietra V.J.D., Pietra S.A.D., Mercer R.L. The mathematics of statistical machine translation: Parameter estimation // Computational linguistics. – 1993. – Vol. 19, №. 2. – P. 263-311.
2. Tiedemann J. Improved sentence alignment for movie subtitles // Proceedings of RANLP. – 2007. – Vol 7. – P – 10-26.
3. Itamar E., Itai A. Using Movie Subtitles for Creating a Large-Scale Bilingual Corpora // Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08). – Marrakech: European Language Resources Association (ELRA), 2008. – P. 209-222.
4. Tiedemann J. Parallel Data, Tools and Interfaces in OPUS // Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'12). – Istanbul: European Language Resources Association (ELRA), 2012. – Vol. 2012. – P. 2214-2218.
5. Mohammadi M., GhasemAghaee N. Building bilingual parallel corpora based on Wikipedia // IEEE Computer Society. – Los Alamitos, 2010. – Vol. 2. – P. 264–268.
6. Weaver W. Translation // Machine translation of languages. – 1955. – Vol. 14. – P. 15-23
7. Britz D., Goldie A., Luong M.T., Le Q. Massive exploration of neural machine translation architectures. 2017. E-print: arXiv:1703.03906 Last accessed: 11 January 2019
8. Klein G. et al. OpenNMT: Open-source toolkit for neural machine translation. 2017. E-print: arXiv:1701.02810. Last accessed: 27 December 2018
9. Sutskever I., Vinyals O., Le Q.V. Sequence to sequence learning with neural networks // Advances in neural information processing systems. – 2014. –Vol. 4. – P. 104-3112.
10. Vorontsov K., Potapenko A. Additive regularization of topic models //Machine Learning. – 2015. – Vol. 101, №. 1-3. – P. 303-323.
11. Papineni K., Roukos S., Ward T., Zhu W.J. BLEU: a method for automatic evaluation of machine translation // Proceedings of the 40th annual meeting on association for computational linguistics. – Philadelphia, Pennsylvania: Association for Computational Linguistics, 2002. – P. 311-318.
12. Koehn P., Hoang H., Birch A., Callison-Burch C., Federico M. Open source toolkit for statistical machine translation: Factored translation models and confusion network decoding // Final Report of the 2006 JHU Summer Workshop. – Baltimore, Maryland: – 2006.
13. Green S., Cer D., Manning C. Phrasal: A toolkit for new directions in statistical machine translation // Proceedings of the Ninth Workshop on Statistical Machine Translation – Baltimore, Maryland, 2014. – P. 114-121.
14. Krasnov F., Ushmaev O. Exploration of hidden research directions in oil and gas industry via full text analysis of OnePetro Digital Library // International Journal of Open Information Technologies. – 2018. – Vol. 6, №. 5. – P. 7-14.
15. Monroe W., Hu J., Jong A., Potts C. Generating bilingual pragmatic color references. 2018. E-print: aXiv:1803.03917. Last accessed: 24 December 2018

Материал поступил в редакцию 08.04.19.

Сведения об авторах

КРАСНОВ Федор Владимирович – кандидат технических наук, ведущий эксперт Газпромнефть НТЦ, Санкт-Петербург
e-mail: Krasnov.FV@gazpromneft-ntc.ru
ORCID: 0000-0002-9881-7371

ШВАРЦМАН Михаил Ефремович – заместитель директора Национального электронно-информационного консорциума, начальник отдела Российской государственной библиотеки, Москва
e-mail: shvar@neicon.ru
ORCID: 0000-0003-2524-6819

ДИМЕНТОВ Александр Владимирович – эксперт Национального электронно-информационного консорциума
e-mail: adimentov@neicon.ru
ORCID: 0000-0002-0357-8495

СЕНЬ Анастасия Игоревна – бакалавр Санкт-Петербургского государственного университета, Факультет прикладной математики - процессов управления
e-mail: anastasiia.sen@apmath.spbgu.ru
ORCID: 0000-0002-1949-1642

УДК 004.91:003.07

Е.Т. Петрова, Т.Г. Петров, С.В. Чебанов, С.В. Мошкин

Метод кодирования многокомпонентных объектов (RHA) и его применение для упорядочения шрифтов прямого начертания

На базе информационного языка RHA, разработанного для разных типов объектов, предлагается систематизировать каркасы начертаний знакотипов "Н" разных шрифтов, с назначением каркасу его кода, посредством помещения знакотипа "Н" в стандартизированное окно (ФонтОкно), и последующим расчётом количественных характеристик (характеризаций). Совокупность кодов визуализируется на диаграмме. Коды упорядочивают по специальному алфавиту SBCO, где S, B, C, O являются, с одной стороны, элементами кода а, с другой – обозначениями компонентов каркаса начертаний, а также по алфавитам: 1) рейтингов распределения полей компонентов в ФонтОкне, 2) энтропий, 3) анэнтропий. Коды сопровождаются названиями закодированных шрифтов. Предлагаемый принцип расположения кодов в списке не зависит от названия, стиля, ширины, назначения шрифта и его автора. Описываемый метод RHA группирует "Н" по похожести их каркасов. Создаваемый список-каталог может включать все латинские и кириллические "Н" прямого начертания. Макет каталога включает 99 кодов каркасов начертаний знакотипа "Н", распределённых по 6 классам из числа 24-х возможных в принятой авторами системе. Предлагаемый вариант применения метода RHA может быть основой создания общего способа кодирования и систематизации как двуцветных, так и многоцветных рисунков на плоскости, включая карты и иные изображения, в состав которых входят наборы компонентов разных цветов и форм.

Ключевые слова: *знак шрифта, аллограф, ФонтОкно, каркас буквы, алфавитное упорядочение кодов шрифта, энтропия каркаса буквы, анэнтропия каркаса буквы, RHA-код каркаса знакотипа., RHA-система каркасов знакотипов, R-класс каркасов знакотипа, энтропийно-анэнтропийная диаграмма, шрифты-синонимы, шрифт-клон, географические карты, геологические карты, многоцветные изображения*

ВВЕДЕНИЕ

Шрифтовое разнообразие, возникающее как результат особого вида художественного творчества, представлено многочисленными вариантами, и каждый год шрифтовые и дизайн-студии выпускают новые. Уже насчитывается свыше 13 тыс. шрифтов. Дизайнеры, авторы и полиграфисты часто сталкиваются с необходимостью поиска шрифтов нужного рисунка, с сортировкой шрифтовой базы на компьютере, на сайте продавца или производителя шрифтов. Существуют пиратские шрифты-клоны, полностью копирующие рисунок авторских версий под изменёнными названиями [1, 2], шрифты-синонимы, созданные по образу какого-либо базового шрифта и похожие на базовый, а также шрифты с одинаковыми названиями, но разным рисунком.

Имеется несколько способов классификации шрифтов [3], из которых можно выделить группирования по: историческим аспектам [4, 5], конструктивным особенностям (с засечками, без засечек [6, 7]), типу создания (рукописные, трафаретные и т.п. [8, 9]), назначению (наборные, акцидентные, символьные [10–12]). Все эти способы упорядочивания шрифтов требуют экспертной оценки. Универсальным (на сегодняшний день) является простое упорядочение шрифтов с помощью естественного алфавита, что используется и в программах (например, *FontExplorer*, *FontExpert*), и в печатных каталогах [13, 14], и на шрифтовых сайтах [15–17]. Однако названия (имена) шрифтов не несут смысловой нагрузки, а потому при их использовании, не зная шрифт «в лицо», не имея измеримых характеристик шрифтов и системы их упорядочения, найти похожие шрифты можно только с помощью специальных приложений [18]. Одно из популярных приложений [19] распознаёт изолированные символы латинского шрифта, подаваемого на вход программы и предлагает на выходе несколько вариантов названий шрифтов на выбор. Сайт *Identifont*, предлагая поиск нужного шрифта средствами анкетирования, сужает поиск до нескольких названий шрифтов [20]. Эти ресурсы доступны и просты в использовании, но разобщены и не дают однозначного упорядочения шрифтов согласно их форме среди подобных по на-

чертанию, не имеют способа прослеживания изменчивости формы от шрифта к шрифту. В эти приложения не входят нелицензионные шрифты [18]. С практической точки зрения существующее положение вполне приемлемо для дизайнеров, занимающихся шрифтами, однако невозможно увидеть общие принципы организации многообразия шрифтов, тенденции их появления, специфику их использования, особенности психофизиологии восприятия, а также предсказывать появление новых шрифтов с определёнными характеристиками.

В [21] отмечено, что построение всеохватывающей детальной классификации шрифтов вообще вряд ли возможно, так как они – продукт художественного творчества. При этом, при их описании гуманитарии обычно стремятся охватить объект во всех деталях. Однако, если не требовать учета ***всех*** особенностей описываемых объектов, может быть, плавно изменяющихся и не имеющих определённых границ между явно различающимися, эти объекты можно описать, например, если выделить какие-то измеримые свойства, важные для данной группы объектов и сравнивать объекты по этим свойствам, как делают естественники (точнее, речь идёт о разных способах работы: с идиографией – выявлением разнообразия и номотетикой – выявлением законов; пример их объединения работа [22]).

В качестве универсальной характеристики непрерывно изменяющихся составов объектов был выбран сдвоенный перечень названий компонентов, слагающих объект, и долей этих компонентов в объекте, который подвергается энтропийно-анэнтропийному анализу (метод ***RHA*** Т.Г. Петрова [23]). Анализ заключается в том, что после получения ранговой формулы ***R***, осуществляется вычисление информационной энтропии К. Шеннона ($\boldsymbol{H}=-\Sigma \boldsymbol{p}_i * \ln \boldsymbol{p}_i$ где $\boldsymbol{p}_i$ – нормированная к 1 частота *i*-ого компонента [24, с. 48]), характеризующей равномерность распределения компонентов, и анэнтропии ($\boldsymbol{A}=-[(\Sigma \ln \boldsymbol{p}_i)/\boldsymbol{n}] - \ln(\boldsymbol{n})$, где $\boldsymbol{n}$ – число компонентов, представляющих объект [24, с. 61]), введённой Т.Г.Петровым для характеристики неравномерности вкладов компонентов в их распределение, на основании сопоставления которых у изучаемых объектов делаются содержательные выводы. При этом ***H*** и ***A*** могут рассчитываться для полных (что позволяет полнее представлять их индивидуальность) или усечённых составов объектов (с целью их сопоставления вне зависимости от природы объектов и способов их изучения).

Для единообразного описания подобного рода представлений составов был разработан информационный язык-метод *RHA* [23], который изначально был предложен для описания химических составов горных пород, а позднее проявил себя как способ кодирования и алфавитного упорядочения составов объектов любой природы, типы компонентов которых либо дискретны, либо дискретизированы [25, 26]. Современное изложение метода *RHA* приведено в [24–31]. К представлению составов (минералов, горных пород, текстов, возрастов населения, видовому составу планктона, типов корреспонденции и т.д.), обрабатываемых по этому методу, предъявляются два основных требования: 1) определённость выделяемых компонентов состава, т.е. однозначность их различения и 2) близость суммы размеров выделенных частей к 100% (или ***к единице***). Точное равенство суммы содержаний компонентов ста процентам скорее вызывает подозрение в качестве анализа, чем свидетельствует о его точности. Что касается оговорки, приведённой в скобках, то следует отметить, что в работах по информатике, к которой принадлежит метод *RHA*, в качестве мер интенсивности свойства или частоты событий используются не проценты, а доли $\boldsymbol{p}_i$ целого, принимаемого за единицу ($\sum \boldsymbol{p}_{ii}=1$).

Итак, *RHA* – это:

- способ представления данных о составе;
- метод первичной обработки этих данных;
- информационно-поисковый язык описания составов;
- способ кодирования данных о составе конкретного объекта;
- код состава данного объекта;
- способ упорядочивания составов.

Имея в виду разнообразие реализованных возможностей метода *RHA*, стало естественным желание продолжить разработку способа на примере кодирования типографских шрифтов, что и было начато в статье [32].

Задачи настоящей статьи – описать буквенно-цифровой способ кодирования символов шрифтов прямого начертания, позволяющий кодировать и упорядочивать их по рангово-энтропийным характеризациям[1] в виде таблицы, представить макет каталога знакотипов "Н" и визуализировать их распределение на энтропийно-анэнтропийных диаграммах.

ЗНАКОТИП «Н» И СТАНДАРТ ЕГО ОПИСАНИЯ

Каркас знакотипа "Н" (рис. 1*а*) понимается как схематическое изображение знака "Н", состоящее из двух вертикальных отрезков сравнимой длины – штамбов и соединительного горизонтального или слабоправонаклонного соединительного отрезка (при левонаклонном соединительном отрезке начинаются знакотипы "И", а при правонаклонном – "N"), высота расположения которого во внимание не принимается (за исключением запрета занятия крайнего верхнего и нижнего положений). Данный знакотип как общий для букв "Н" латиницы и "Н" кириллицы был выбран в статье [32] по этой теме. Этот знакотип, будучи в конкретном шрифте художественным произведением, может иметь многочисленные особенности, которые при формализованном описании шрифта не принимаются во внимание. Так, игнорируются разнообразные такого рода модификации перекладины и штамбов, также как и неравномерности тона или толщины компонентов знакотипа, и мы получаем

[1] В отличие от неколичественных характеристик объектов, в дальнейшем *количественные характеристики*, в нашем случае – энтропию и анэнтропию – будем называть *характеризациями*.

упрощенное изображение знакотипа «Н» – ***каркас знакотипа*** «***Н***», который и подлежит кодированию (рис. 1). В результате будет получен код «Н» конкретного шрифта той или иной гарнитуры.

Для сопоставимости разных каркасов их высота принимается за единицу, а для сравнимости ширины каркасов введена максимальная возможная ширина, за которую принята сумма единицы и числа φ золотой пропорции, т. е. значение 1+1.618 = 2.618. Добавление единицы обусловлено тем, что существуют варианты "Н", ширина которых превышает 1.618 при высоте, равной единице. Для визуального представления и сравнимости каркасов "Н" разных шрифтов строится специальное окно – «***ФонтОкно***» (F) – с фиксированной высотой, принятой за единицу, и шириной равной 2.618. В качестве примера полученный каркас "Н" шрифта *Braggadoci* помещается в ФонтОкно (рис. 2).

ФонтОкно площадью F_S с помещённым в неё каркасом, имеет компоненты, для которых устанавливаем следующие обозначения:

1) два основных вертикальных штриха каркаса, имеющие в сумме площадь S_S (*Stem);*

2) соединительный горизонтальный штрих каркаса с площадью B_S (*Bar*);

3) внутреннее пространство каркаса с площадью C_S (*Counter*);

4) остаточное свободное пространство с площадью O_S (*Outdoor*).

Сумма площадей выделенных компонентов $F_S = S_S + B_S + C_S + O_S = 2{,}618$, что принимается за 100%, а при дальнейших расчётах за 1.

Рассмотрим идеализированные случаи форм знака шрифта при приближении выделенных параметров к 100% площади S_F ФонтОкна F (рис. 3).

a) *b*) *c*)

Рис. 1. Переходы от знакотипа (*a*) к аллографу «Н» шрифта *Braggadocio* (*b*) и каркасу «Н» шрифта *Braggadocio* (*c*).

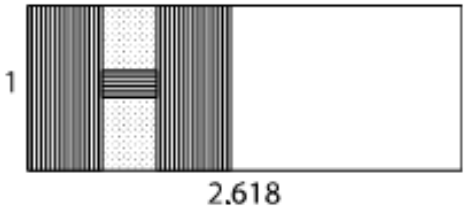


Рис. 2. ФонтОкно с вписанным в него каркасом "Н" ,

где:

S – штамбы
B – соединительный штрих
C – внутреннее пространство аллографа
O – остаточное пространство ФонтОкна.

Рис. 3. Крайние – идеализированные формы знака шрифта "Н":

a) так выглядит "Н" при суммарной площади штамбов – S_S, приближенной к 100%. Чем больше значение S_S, тем меньше площадь внутреннего пространства и шире аллограф, шрифт насыщеннее;

b) если площадь B_S (площадь соединительного штриха) приближается к 100%, в ФонтОкне остаётся перекладина с исчезающим участием всех остальных компонентов аллографа;

c) приближение к 100% внутреннего пространства аллографа (C_S) влечет приближение к нулевым площадям остальных компонентов в ФонтОкне, что влияет на светлоту буквы;

d) при свободной площади O_S, приближающейся к 100%, будем иметь аллограф с исчезновением всех остальных частей. Аллограф получается сверхузким.

Итак, получаем набор площадей компонентов каркаса буквы и площадь свободного пространства в ФонтОкне. Этот набор предстоит преобразовать в код по методу *RHA*.

ПРЕДСТАВЛЕНИЕ ФОНТОКНА С КАРКАСОМ "Н" НА ЯЗЫКЕ *RHA*

Ранговая формула *R*. Описание знакотипов на языке *RHA* включает полуколичественную характеристику знакотипа по составу его компонентов с помощью ранговой формулы ***R*** [23] (первый символ в названии метода) и количественное описание площадного состава знакотипа. Ранжирование значимостей при описании совокупностей частей – это весьма распространённая процедура, но такие последовательности до сих пор в науке не рассматривались, поскольку не осознавались как особый объект. Однако ранговые формулы являются принципиально новым объектом, представляющим интерес для информатики. Ранговая формула ***R*** компонентного состава знакотипа "Н" как первая часть кода – это последовательность имён компонентов каркаса по снижению долей их площадей в ФонтОкне. Таким образом, ранговая формула содержит два вида данных о каркасе: 1) качественная характеристика о нём, так как в ней сообщается о наборе конкретных компонентов каркаса, находящегося в ФонтОкне; 2) порядковые характеризации, так как в ранговой формуле сообщается об относительных величинах компонентов: важности, значимости, размере площадей деталей каркаса в ФонтОкне.

Построим ранговую формулу для каркаса "Н" шрифта *Braggadocio* (см. рис. 1 и *Приложение*[2], №32). Каркас "Н" этого шрифта имеет следующие площади деталей (в %): S_S =42.77, B_S =0.30, C_S =3.20, O_S =53.73. Записывая площади с их аббревиатурами по снижению значений и используя знаки неравенств, получаем последовательность **O>S>C>B**. После этого, освобождаясь от знаков неравенства и значений площадей, получаем ранговую формулу $\mathbf{O_S S_S C_S B_S}$. Другой пример – каркас шрифта *Distill* (см. *Приложение*, №66): S_S =7.84, B_S =4.39, C_S =36.50, O_S =51.27 – ему соответствует ранговая формула $\mathbf{O_S C_S S_S B_S}$.

Таким образом, предложенное ранжирование площадей разбивает всё возможное разнообразие прямых шрифтов на группы, имеющие одинаковые ранговые формулы. Это первый шаг в упорядочении многообразия шрифтов. Одной и той же ранговой формуле может соответствовать неопределённо большое количество конкретных сочетаний площадей. Среди них встречаются формулы, имеющие соседние компоненты, практически не различающиеся по содержаниям. Для выделения таких компонентов используются знаки равенства.

Не существует каких-то определённых границ между выделяемыми в языке степенями сходства от "очень" похожего (две книги одного тиража) до "почти ничего общего" (две ноги одного человека и крик чайки). С другой стороны, в некоторых областях знания (кристаллохимия, минералогия) оказывается оправданным рассматривать величины, различающиеся не более чем на 15 относительных[3] процентов как равные.

Чтобы не искать других обоснований и не порождать уводящих в сторону обсуждений, примем эту границу и в нашем случае. Соответственно, будем ставить знак равенства между соседними символами в ранговой формуле площадей каркаса, если результат деления предыдущей (в ранговой формуле) доли площади на последующую не превышает 1.15. При этом порядок компонентов – деталей каркаса – в ранговой формуле сохраняется, и только при точном равенстве порядок этих символов задаётся алфавитом (S B C O).

Вторая часть кода состава, вводимая после ранговой формулы, – это интегральная характеризация каркаса, энтропия (***H***) – ответственна ***главным образом*** за средне-большие доли площади, третья, анэнтропия (***A***), – ***главным образом*** за малые.

Энтропия и анэнтропия позволяют более полно учесть разнообразие набора размеров площадей. Обратимся к этим понятиям.

Энтропия *H*. Рассмотрим изучаемый комплекс (ФонтОкно со вписанным в него каркасом "Н") с точки зрения степени *равно*размерности площадей его компонентов: S_S,B_S,C_S,O_S. Экстремально равноразмерный комплекс представлен каркасом, вписанным в ФонтОкно, в котором площади всех компонентов содержимого окна равны друг другу.

Экстремально *разно*размерное содержимое окна заполняется одной-единственной «любой» деталью (рис. 3). Соответственно, количество экстремально *разно*размерных комплексов будет равно числу компонентов комплекса (т.е. 4), а предельно равноразмерный комплекс будет единственным, что может пригодиться в дальнейшем.

В качестве меры равномерности примем информационную энтропию К. Шеннона, которая используется во многих отраслях знания [33]. На стадии зарождения метода RHA она была предложена для геохимии в качестве меры сложности химического состава[4] [34]. Она определяется формулой:

$$\boldsymbol{H} = -\sum \boldsymbol{p}_i \ln \boldsymbol{p}_i,$$

где $\boldsymbol{p}_i$ – частота события, в данном случае, равная долям площадей компонентов содержимого ФонтОкна, в силу чего $\sum \boldsymbol{p}_i$ =1. В рассматриваемом случае $\boldsymbol{p}_i$ – доля площади, занимаемая *i*-й деталью содержимого ФонтОкна. Величина «$-\boldsymbol{p}_i \ln \boldsymbol{p}_i$» – вклад отдельной детали содержимого ФонтОкна в энтропию (рис. 4). Если $\boldsymbol{p}_i$ = 0, то вклады данного компонента в энтропию не рассчитываются. При $\boldsymbol{p}_i$ = 1 вклад компонента равен 0.

[2] Здесь и далее, для того чтобы продемонстрировать примеры знакотипов "Н" разных шрифтов, будут даваться ссылки на *Приложение* до того, как будут описаны принципы его построения.

[3] Под относительными процентами понимается отношение двух величин, выраженное в процентах при делении большей на меньшую.

[4] Использование информационной энтропии как меры сложности, в принципе, не единственный вариант оценки сложности (например, решение сложной задачи энтропией не оценивается).

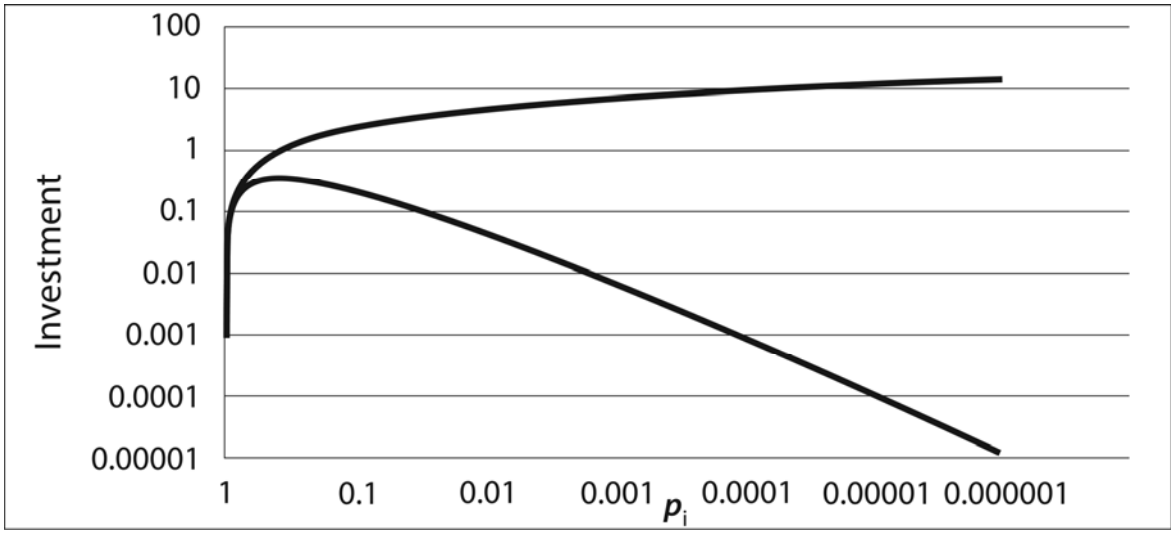


Рис. 4. Зависимости активных вкладов (*Investment*) в энтропию «$-p_i\ln p_i$» – (нижняя кривая) и в анэнтропию «$-\ln p_i$» (верхняя кривая) от долей площадей p_i

Как видим на рис. 4, зависимости активных[5] вкладов в энтропию и анэнтропию от содержаний до достижения максимума вклада в энтропию при p= 0.368... ведут себя симбатно, после достижения указанного максимума – противоположным образом.

При равномерном распределении p, т.е. при p_1= p_2= p_3 =...= p_n=1/n энтропия состава объекта, которую можно интерпретировать как его сложность, является максимальной и равна ln$n$ (логарифмы натуральные), где n – число учтённых компонентов ФонтОкна. Так заполненное ФонтОкно будем называть *равно*размерным. В нашем примере энтропия максимальна и равна ln4=1.386... Такой каркас знакотипа "H" был нами построен и изображен на рис. 5. Шрифт, разработанный на базе такого образца, мог бы получить название **SuperCompl*[6] – предельно равноразмерный.

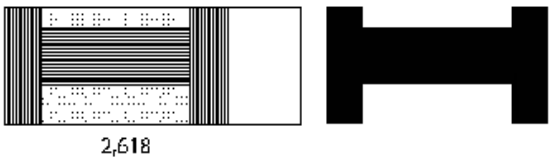


Рис. 5. Компоненты «H» при условии равенства всех площадей в ФонтОкне.

Объект, состоящий из одной детали (p=1), имеет минимальную равноразмерность равную 0. Применительно к анализу каркаса, когда равноразмерность минимальна, это соответствует пустому или равномерно заполненному одной деталью (компонентом) ФонтОкна (примеры – рис. 3*a*,*b*,*c*,*d*). В нашем случае, если свободное поле занято одной-единственной деталью то, *как и многие другие идеалы, этот идеал простоты обессмысливает использование такого знакотипа.* К нему наиболее близок каркас "H" шрифта *Super C* (*Приложение,*№64) практически полностью представленный свободным пространством. На грани с такой ситуацией находятся так называемые «концептуальные шрифты», использующие отказ от некоторых элементов буквы, например, "H" шрифта *Gropius Display* (*Приложение*, №10), у которого в каркасе «H» отсутствуют перекладина и свободное пространство между штамбами, а также *GHSans* (*Приложение*, №22), «H» которого лишен перекладины. Первый имеет ранговую формулу $O_S S_S B_S$=C_S и энтропию H=0.665, второй – $O_S S_S C_S B_S$ и H=0.866.

Анэнтропия *A*. Третья интегральная характеризация состава – анэнтропия – была предложена Т.Г. Петровым [23] для уменьшения степени неопределённости описания состава многокомпонентной системы при её отображении двумя характеризациями – ранговой формулой и энтропией. Понятно, что чем больше компонентов в системе и меньше её характеризаций, тем больше недоопределённость сведений о конкретном объекте, системе, организме и т.д.

Анэнтропия рассчитывается по формуле

$$A = -1/n\sum \ln p_i - \ln n,$$

где обозначения те же, что и в формуле энтропии. Активным вкладом p_i компонента в анэнтропию является «$-\ln p_i$».

[5] Как активные квалифицируются вклады, которые напрямую зависят от присутствия рассматриваемых компонентов и не содержат в явном виде параметров, связанных с процедурой расчёта (количеством компонентов). Они противопоставляются пассивным вкладам, расчёт величины которых в явном виде зависит от параметров расчёта. В нашем случае параметр пассивного вклада в формулах равен количеству компонентов – четырём.

[6] Знак "*" ставится перед несуществующим, но придуманным в качестве мысленного эксперимента лингвистическим (семиотическим) объектом.

Анэнтропия, определяемая средним арифметическим активных вкладов, взятых с отрицательным знаком, относительно слабо зависит от больших $\boldsymbol{p}_i$ и сильно зависит от малых. Она изменяется от 0 – при равномерном распределении, т.е. когда энтропия максимальна, до неопределённо большой величины – при приближении к нулю хотя бы одного $\boldsymbol{p}_i$. Для шрифтов последняя ситуация очень редка, но, чтобы не отказывать им в упорядочивании, все характеризации их знакотипов приводятся в *Приложении*. На диаграммах же их можно помещать над значением энтропии, вплотную к верхнему краю диаграммы, несколько нарушая линию рамки, чтобы отметить аномальность точки и акцентировать внимание на том, что при стремлении $\boldsymbol{p}_i$ к нулю анэнтропия стремится к $+\infty$. Например, это относится к аномальным знакотипам шрифтов *Gropius Display* №10, *GHSans* №22,), почти перестающим быть буквами, или к фигурам, которые изображены на рис. 3, где *a)* – приближаются к 0 доли деталей площадей B, C и O; *b)* – исчезающие площади S_S, B_S, O_S; *c)* – приближающиеся к нулю площади S_S, C_S, O_S; *d)* – стремятся к нулю площади S_S, B_S, C_S.

Итак, анэнтропия рассчитывается для тех же площадей в ФонтОкне, что и энтропия, но, в отличие от неё, в качестве вкладов используется отрицательный логарифм доли площади, т.е. «$-\ln \boldsymbol{p}_i$». Эта величина приближается к 0 при приближении $\boldsymbol{p}_i$ к 1 (и это для всех оснований логарифмов), т. е. к площади всего ФонтОкна, и монотонно возрастает при уменьшении $\boldsymbol{p}_i$.

СПОСОБ УПОРЯДОЧЕНИЯ КОДОВ-СЛОВ *RHA* ЗНАКОТИПОВ "H"

Совокупность ранговой формулы, энтропии и анэнтропии рассматривается как код ФонтОкна с его содержимым и, одновременно, как слово в информационном языке *RHA*. Для построении этого слова привлечено три алфавита.

Первый алфавит – это задаваемый авторами статьи порядок символов компонентов: S C B O, приведённый в обозначениях рис. 2. Эта последовательность принята нами за алфавит для упорядочения ранговых формул каркасов букв. Сначала вся совокупность конкретных кодов-слов *RHA* ФонтОкон (в сопровождении названий шрифтов) разбивается на группы с одинаковыми первыми членами ранговых формул, после чего группы располагаются в соответствии с алфавитом (SBCO), как и при использовании алфавитов существующих письменностей. Затем внутри каждой группы операция повторяется со вторым членом ранговой формулы и так далее. В результате для четырехбуквенной (рейтинговой) части кодов имеем полный набор возможных ранговых формул, которые в таблице расположены как слова в алфавитной последовательности по **SBCO,** где полужирным шрифтом выделены ранговые формулы, для которых существуют реализации.

Второй алфавит континуальный – величины энтропии. В пределах группы одинаковых ранговых формул для алфавитного упорядочивания знакотипов используется энтропия, выражаемая рациональными числами. Однако, направление упорядочивания описаний объектов в пределах группы с одинаковыми ранговыми формулами, в принципе, может быть разным. Так, в статье об алфавитах [32] использовалось расположение кодов по неубыванию $\boldsymbol{H}$, как и в случае упорядочения кодов возрастного распределения населения [26]. В статье [32] направление упорядочивания было принято без каких-либо обоснований, в [26] направление оправдывалось историко-демографическими причинами. По мере накопления материала по "H", и анализа диаграмм $\boldsymbol{HA}$, стало ясно, что использованный в [32] вариант упорядочения шрифтов затрудняет интерпретацию результатов. Поэтому было принято решение об алфавитном упорядочении значений $\boldsymbol{H}$ по их невозрастанию[7], что поясним далее.

Все возможные ранговые формулы – ***R*-**классы знакотипов "H",
упорядоченные по алфавиту SBCO

1	$\mathbf{S_sB_sC_sO_s}$	7	$B_sS_sC_sO_s$	13	$C_sS_sB_sO_s$	19	$\mathbf{O_sS_sB_sC_s}$
2	$S_sB_sO_sC_s$	8	$B_sS_sO_sC_s$	14	$C_sS_sO_sB_s$	20	$\mathbf{O_sS_sC_sB_s}$
3	$S_sC_sB_sO_s$	9	$B_sC_sS_sO_s$	15	$C_sB_sS_sO_s$	21	$O_sB_sS_sC_s$
4	$S_sC_sO_sB_s$	10	$B_sC_sO_sS_s$	16	$C_sB_sO_sS_s$	22	$O_sB_sC_sS_s$
5	$S_sO_sB_sC_s$	11	$B_sO_sS_sC_s$	17	$\mathbf{C_sO_sS_sB_s}$	23	$\mathbf{O_sC_sS_sB_s}$
6	$\mathbf{S_sO_sC_sB_s}$	12	$B_sO_sC_sS_s$	18	$C_sO_sB_sS_s$	24	$O_sC_sB_sS_s$

[7] Имеется в виду, что разные шрифты могут иметь равные значения энтропии.

Третий алфавит также континуальный – величины анэнтропии. Для упорядочивания кодов с одинаковыми ***R*** и ***H***, используются значения ***A*** по их неуменьшению, которое было выбрано в связи с тем, что статистически преобладает обратная зависимость между ***H*** и ***A***.

RHA-УПОРЯДОЧИВАНИЕ ЗНАКОТИПОВ "H"

Коды 99 произвольно отобранных шрифтов латиницы из коллекций ПараТайп, *MyFonts*, *FontShop* представлены в *Приложении*. Ранговые формулы упорядочены по алфавиту SBCO. Энтропии в пределах одной ранговой формулы – ***R***-класса шрифта – упорядочены по невозрастанию. В колонке "Каркас в F" каркасы вставлены в ФонтОкно так, что левая граница ФонтОкна совпадает с их левой границей. Это позволяет сделать следующие выводы:

1) из общего количества – 24 – возможных ***R***-классов "H" (см. таблицу) представлены 6 классов (включая первый – предложенный только в настоящей статье);

2) около 90% всех учтённых авторами в настоящей статье знакотипов "H" относятся всего к двум ***R***-классам:

$O_sS_sC_sB_s$ (№№ 11-64) внешнее пространство – штамбы – соединительный штрих – внутреннее пространство,

$O_sC_sS_sB_s$ (№№ 65-99) внешнее пространство – внутреннее пространство – штамбы – соединительный штрих.

Преобладает первый класс – "H", в котором 54 шрифта – с относительно толстыми штамбами и второй, где 35 – более "тощих";

3) теоретическое построение каркаса "H", названного нами **SuperCompl*, с ранговой формулой $S_sB_sC_sO_s$, с последовательностью символов, совпадающей с последовательностью символов в алфавите, максимально возможной (для 4-х компонентов) энтропией и минимально возможной анэнтропией, определяет роль этого каркаса как начала для упорядочения шрифтов по ранговым формулам (см. таблицу), энтропии (см. *Приложение*) и анэнтропии. Последовательности символов в начале слов в алфавитных словарях максимально похожи на последовательности символов в начале алфавита, в идеале – должны быть самим алфавитом. Соответственно, последний ***R***-компонент слова ***RHA*** (ранговая формула) будет представлять последовательность символов обратную алфавиту;

4) расположение теоретического каркаса "H" **SuperCompl* в качестве первого при упорядочении определяется способом упорядочения строк таблицы и внутри ***R***-классов. Его исключительность следует из того, что: а) этот знак содержит все компоненты, хорошо визуально различимые; б) это изображение, обладая максимальной энтропией, при любых изменениях пропорций между четырьмя компонентами, будет трансформироваться в любые разные, но имеющие меньшую энтропию, как в классе $O_sS_sB_sC_s$, к которому он приписан в макете каталога, так и во всех других, возникающих при изменении исходного равенства площадей компонентов ФонтОкна.

Обратим ещё раз внимание на то, что при устремлении доли площади любого компонента к 1 до практической нераспознаваемости каркаса будет возникать тот же эффект, что и при заливке ФонтОкна одним тоном, т. е. энтропия будет уменьшаться до нуля, а анэнтропия – бесконечно расти. При устремлении к 0 площадей одного или двух компонентов останутся два или три компонента; ***A*** будет равна $+\infty$;

5) в последовательностях по невозрастанию энтропии наиболее широко представленных классов каркасов $O_sS_sC_sB_s$ и $O_sC_sS_sB_s$. (на фоне общего уменьшения зрительно воспринимаемой ширины каркаса вплоть до каркаса шрифтов *Super C* №64 и *FR Pasta Mono* №99, практически неразличимых визуально) каркасы первого класса имеют значительно большее разнообразие величин энтропийных параметров, чем разнообразие у второго. Это следует и из *Приложения* и из представления распределения соответствующих точек на рис. 6;

6) тождественные и наиболее похожие по внешнему виду каркасы имеют преимущественно одинаковые ранговые формулы и одинаковые или близкие энтропийные характеризации. В *Приложении* они находятся по соседству, как например: *PT_PTS*75 *Bold* №25 и *PT_PTS*65 *Demi* №26 (имеющие коды: ***R*** – $O_sS_sC_sB_s$, ***H***= 0.860, ***A***= 0.664 и ***H***= 0.858, ***A***= 0.671, соответственно), *Tahoma* №78 и *Kabel* №79 ($O_sC_sS_sB_s$, 0.837, 0.702 и $O_sC_sS_sB_s$, 0.835, 0.704). или близко, например, Regata № 21 и PT_PTS95 Black № 30 ($O_sS_sC_sB_s$, 0.866, 0.667 и 0.848, 0.713).

Исключения, например, наибольшая близость по начертанию шрифтов 49 и 86, или 52 и 89, связаны с тем, что при близости площадей разных компонентов каркаса малые изменения одной из площадей могут приводить к перестановке в ранговой формуле;

7) визуально, последовательности кодов по изменению ***H*** в двух, наиболее представленных ***R***-группах в таблице обнаруживают некоторые общие тенденции изменения формы каркаса. Эта общность связана с тем, что в ФонтОкнах этих групп преобладает и нарастает доля свободного пространства, поэтому в обоих случаях к концу этих групп каркасы становятся всё более узкими. В случаях, когда на первом месте в ранговой формуле находятся другие компоненты ФонтОкна, увеличение их доли площади приводит к формированию монструозных каркасов, типа изображенных на рис. 3*a*, *b*, *c*;

8) на фоне этих тенденций выделяются каркасы, в той или иной степени резко отличающиеся от принадлежащих общим тенденциям. Это следствие того, что каркас имеет три характеризации (***R, H, A,***) при условии $\sum p_i=1$, а при линейном упорядочении описаний используются одна. Отклонения от тенденции изменения форм каркасов в *Приложении*, особенно резко выраженные в блоке $O_sS_sC_sB_s$, – отражение недостаточности представления составов каркасов, имеющих более двух компонентов, одной-единственной информационной энтропией. В качестве примеров каркасов, энтропия которых не отличается или отличается не более чем на 0,001, но существенно различается анэнтропия, приведём №№ 17-18, 31-32, 34-35, 43-44, 46-47, 48-49. При близости значений энтропий и анэнтропий, различия между кар-

касами становятся незначительными, почти исчезающими: №№ 24-25, 58-59, 74-75, 92-93. В данном случае мы имеем яркую визуальную иллюстрацию положения порой о необозримой широте разнообразия того, что оценивается единственным числом, когда бывает порой очень значимо разнообразие значений.

Вопрос об отклонениях от тенденций изменения каркасов при линейном упорядочении их энтропии в большой степени снимается использованием приводимых далее энтропийно-анэнтропийных диаграмм.

КАРКАСЫ ШРИФТОВ НА ЭНТРОПИЙНО-АНЭНТРОПИЙНОЙ ДИАГРАММЕ *HA*

Для общего обзора энтропийно-анэнтропийных характеризаций коллекции шрифтов приведём рис.6.

Как видим, точки на диаграмме распределены резко неравномерно, с тяготением к сравнительно небольшой области в интервалах H 0.6÷1.0 и A 0.6 ÷ 1.2.

Теоретически построенный по принципу максимизации ***H*** и минимизации ***A*** каркас **SuperCompl*, пока единственный представитель ***R***-класса $S_sB_sC_sO_s$, находится в правом нижнем углу диаграммы. Шрифты, энтропийно-анэнтропийные характеризации которых располагаются правее и ниже его на предложенной диаграмме, не существуют и существовать для каркаса с четырьмя компонентами не могут. В то же время другая, не менее важная, особенность этого каркаса заключается в том, что ничтожные изменения площадей его компонентов влекут порождение всех пяти имеющихся и 18 недопредставленных классов в *Приложении*. И больше пока не обнаружено ни одного класса.

Поскольку переход к **SuperCompl* от любого имеющегося класса каркасов может происходить за счёт малых изменений любого компонента, то отклонения от **SuperCompl* могут продолжаться в неопределённом заранее направлении, порождая новые каркасы. Подобная ситуация имеет место в случае странных аттракторов [35] или описываемых в функциональном анализе бифуркаций [36, 37]. Каркас **SuperCompl* выступает как некий *чистый идеал, утрата полноты которого приводит к появлению всех остальных 23 классов* каркасов, каждый из которых имеет какое-то отклонение от идеального **SuperCompl* за счёт кенозиса (умаления) минимум одного из компонентов и возвышения других. Таким образом, все возможные каркасы в потенции находятся в одном идеальном варианте и являются его эманациями [38]. Такие же отношения существуют между вектором с модулем 0, который никуда не направлен, и сколь угодно малым вектором, который имеет одну и только одну направленность, в то время как все остальные направленности исключены, такова же и геометрическая точка по отношению к любой иной фигуре.

Максимальные значения энтропии и одновременно наиболее близкие к теоретическому значению шрифта **SuperCompl* имеют каркасы двух шрифтов ***R***-класса $S_sO_sC_sB_s$ и по одному из классов $O_sS_sC_sB_s$ и $O_sC_sS_sB_s$. Соотношения площадей четырёх компонентов ФонтОкна таких шрифтов наиболее близки друг к другу и к соотношению теоретического **SuperCompl*.

Минимальные значения энтропии имеют каркасы с ранговыми формулами $O_sS_sC_sB_s$ и $O_sC_sS_sB_s$ – они наиболее узкие и последние в перечнях шрифтов соответствующих ***R***-классов. У этих шрифтов в ФонтОкне резко преобладает свободное пространство.

Рекордно большие значения анэнтропии (***A***=+∞) имеют каркасы "H" концептуальных шрифтов: *Gropius Display* (№10), который имеет лишь два компонента кодирования шрифтов (штамбы и свободное пространство ФонтОкна) и *GHSans* (№22) – имеет три компонента. Они отсутствуют на диаграмме.

Наиболее представленные в системе ***R***-классы $O_sS_sC_sB_s$ и $O_sC_sS_sB_s$, при том, что поля их энтропийно-анэнтропийных характеризаций перекрываются в значительной степени, имеют различия. При равных значениях энтропии каркасы "H" $O_sS_sC_sB_s$ в большинстве случаев имеют более высокие значения анэнтропии (рис. 6).

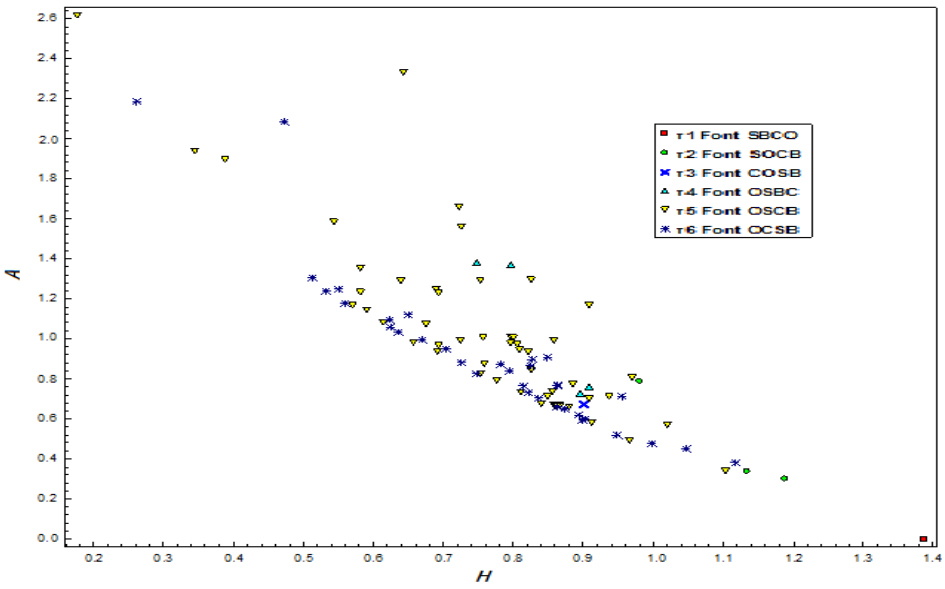


Рис. 6. Энтропийно-анэнтропийные характеризации каркасов, приведённых в *Приложении*

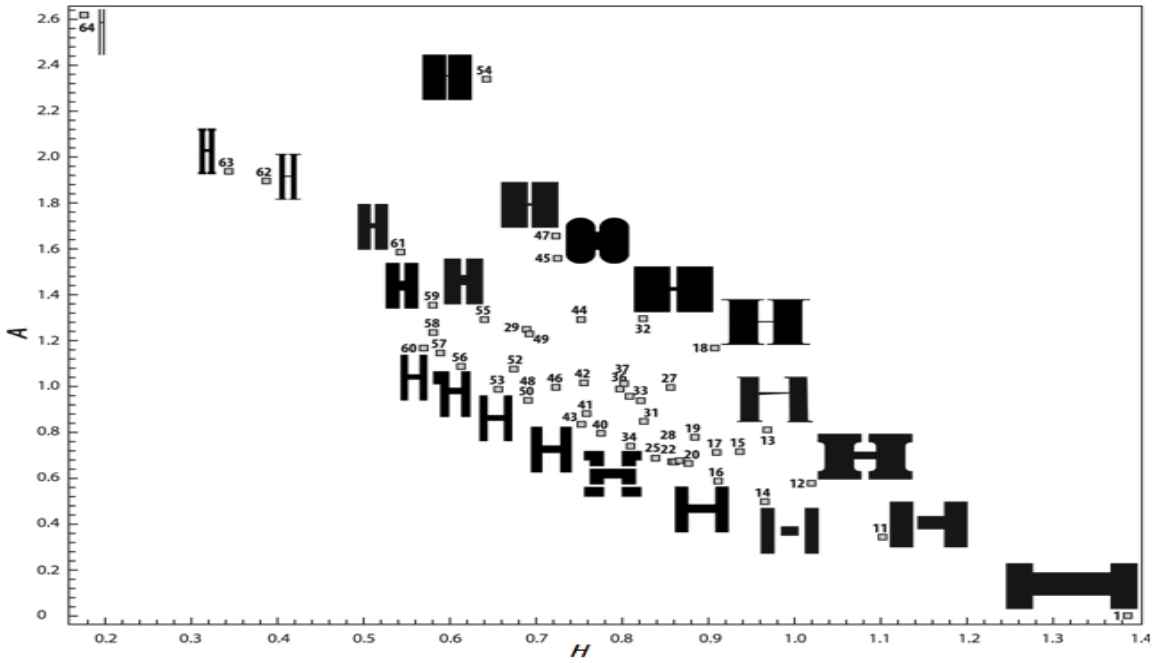


Рис. 7. Аллографы "Н" с ранговой формулой $O_sS_sC_sB_s$ и S_s=B_s=C_s=O_s (№1– в нижнем правом углу) на энтропийно-анэнтропийной диаграмме (номера при точках соответствуют номерам в *Приложении*).

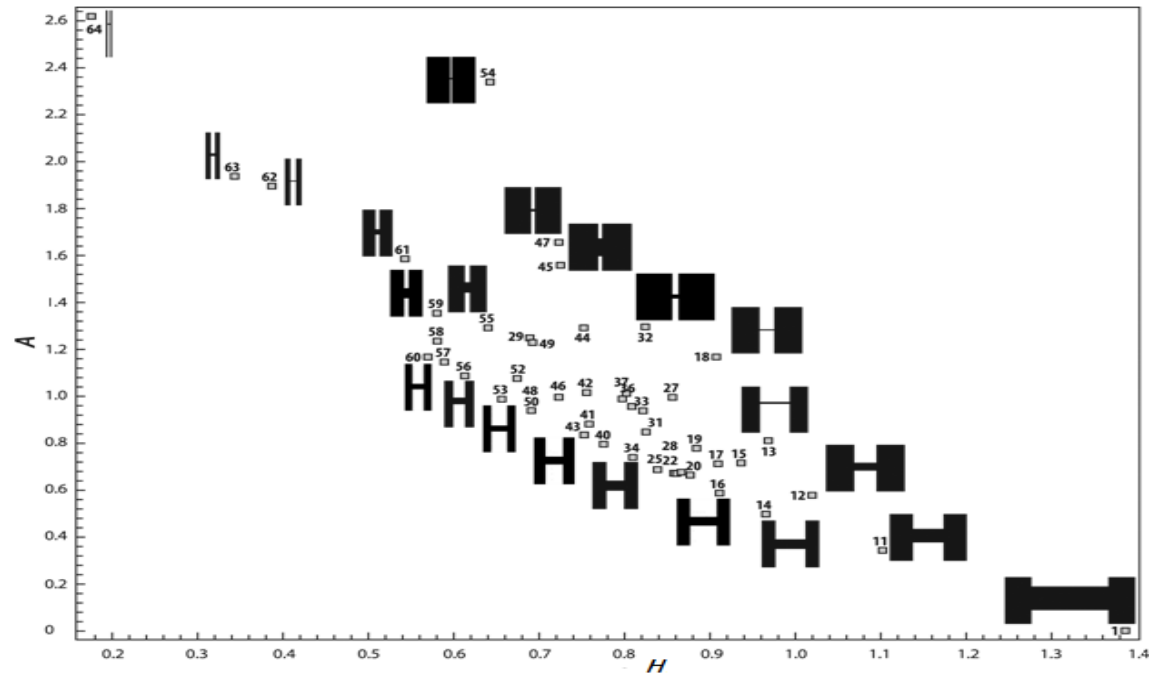


Рис. 8. Каркасы "Н" с ранговой формулой $O_sS_sC_sB_s$ и S_s=B_s=C_s=O_s (№1) на этропийно- анэнтропийной диаграмме.

Высокая плотность точек на диаграмме не позволила сопроводить положение точек изображениями аллографов и их каркасов и сделать картину более наглядной без значительного сокращения количества точек на диаграмме. На рис. 6 показаны ***HA*** и изображения аллографов ***R***-класса $O_sS_sC_sB_s$, на рис. 7 – изображения каркасов того же ***R***-класса.

При сравнении характеризаций ***H*** и ***A*** каркасов "Н" на рис. 7, появляется возможность наблюдать общие тенденции изменения формы каркасов и их группирование.

Каркасы, компоненты которых соизмеримы по площади в ФонтОкне, занимают область больших значений энтропии. Соответственно, каркасы, компоненты которых имеют очень малые доли среди прочих компонентов, имеют большие значения анэнтропии. Два из обнаруженных в литературе варианта: лишенный перекладины *GHSans* (№22) и *Gropius*

Display (№10), лишенный двух компонентов, имеют максимальное значение анэнтропии, равное +∞. Поэтому они не помещены на диаграмму.

Вдоль нижнего края области точек (рис. 7, рис. 8) справа налево – снизу вверх хорошо прослеживается тенденция от широких шрифтов к наиболее узким, что коррелирует с увеличением площади свободного пространства в ФонтОкне.

При равенстве или близости ***R***, ***H*** и ***A*** составы и изображения знакотипов "H", либо одинаковы, либо близки по соотношению площадей каркаса, хотя различаются эстетически.

Коды шрифтов-клонов имеют одинаковые значения с "честными" шрифтами, поэтому клоны на диаграмме будут находиться в одной и той же точке или практически совпадать с ранее появившимися – "честными".

Таким образом, при переходе от аллографа к его каркасу происходит смена выделенностей знакотипов: в случае аллографов выделяются имеющие особенности начертания (№№ 28, 36, 44, 69, 89, 92, 95) и отсутствующие компоненты (№№ 10, 22), а в случае каркасов – имеющие экстремальные энтропийные характеризации (№1 Max***H***,min***A*** и №64 – min***H***Max***A***)

Схематизация кодирования компонентного состава каркасов влечёт резкое сокращение качественного разнообразия, в частности, за счёт исключения, в первую очередь, явно эксцессивных (резко выделяющихся, "патологических") вариантов аллографов.

ОБСУЖДЕНИЕ РЕЗУЛЬТАТОВ

Представленная коллекция ("коллекция" в смысле Фуко [39]), насчитывает 99 шрифтов, то есть она является представителем универсума с неизвестной структурой и пределами варьирования. Она еще мала для общих заключений, поэтому пока можно высказать только некоторые предварительные соображения.

Представленные нами в таблице ранговые формулы исчерпывают своё разнообразие 24 классами, из которых при произвольном выборе 99 шрифтов, представленных в *Приложении*, реализовано всего 5 классов (6-й – сконструирован нами), причём подавляющая их часть приходится только на два класса. Поэтому приходится признать, что используется лишь небольшая часть логически допустимого разнообразия прямых шрифтов.

Большой проблемой был выбор направления упорядочения энтропии и анэнтропии из-за отсутствия очевидных, бесспорных мотивов для определения знака изменений □ к уменьшению или к возраствнию (это направление для горных пород было определено тем, что кристаллизация магм происходит при снижении температуры, а это, в свою очередь, направляет состав магматического раствора к разделению его и статистически достоверному снижению энтропии новых возникающих систем □ минералов по сравнению с исходным силикатным раствором.

Шрифты как произведения искусства, в общем, не связаны друг с другом, и их авторам, видимо, было не известно, как их рисунок развивался во времени, поэтому до построения теоретической формы буквы и набора "достаточно" полной коллекции знакотипов проблема их упорядочения даже не формулировалась. Первично заимствованное из других типов дискурса упорядочение по возрастанию энтропии (как в статье [32]) по мере накопления данных уступило более эвристичному под давлением другой – более высокой согласованности целого при обнаружении новой точки отсчёта. В качестве начальной точки проявил себя каркас "H" **SuperCompl*, занимающий на диаграмме (рис. 6) крайнее нижнее и крайнее правое положение. Этот каркас выступает как порождающая модель [40] всех остальных каркасов.

Малое разнообразие ранговых формул (оно, видимо, несколько возрастёт по мере предполагаемого накопления материалов) говорит о довольно узких информационно-эстетических требованиях, предъявляемых к шрифтам потребителями и обслуживающих их дизайнеров.

При варьировании соотношения площадей каркаса появляется возможность прогнозирования и построения формы каркасов "H", располагающихся между соседними точками на диаграмме ***HA*** или на любой линии, привязанной к этой диаграмме, которую можно будет считать линией семейства каркасов.

Таким образом, даже имеющийся материал показывает, что метод ***RHA*** даёт вариант подхода к вынесению новых суждений о реализованном и потенциальном разнообразии шрифтов.

ЗАКЛЮЧЕНИЕ

Возникает вопрос: Зачем всё это, если уже есть освоенные способы поиска похожих рисунков шрифтов? На это можно дать общий ответ.

Обращение науки к искусству – это средство понять, "как ЭТО делается", "почему лучше ТАКОЕ начертание", "почему ТАК убедительнее, красивее, эмоциональнее, уместнее...". Нельзя ли понять, и это любопытно, почему из 24 классов шрифтов используется меньше трети? А что будет, если сочинить макеты букв – каркасы – пока еще для пустых классов? Может быть, они – эти макеты дадут импульс фантазии художника к созданию чего-то приемлемого, нового?

Для примера такой деятельности был выбран каркас шрифта *Arial* (см. *Приложение*), № 73) и, приняв постоянным набор его метрических характеризаций (17.56, 2.33, 10.09, 70.02), произвели перестановки их значений для компонентов новых каркасов. Так были построены два каркаса с новыми ранговыми формулами, а именно, *COBS* [redacted] и *SBOC* [redacted], а также один – с той же ранговой формулой, что и пока единственный теоретический (**SuperCompl*) *SBCO* [redacted]. Если первые два представляются мало перспективными, то последний вариант, как мы полагаем, можно считать вполне приличным для использования в неких значительных текстах заголовков-надписей.

Между характеризациями букв – полными "словами" на языке *RHA* – и обычно выделяющимися свойствами букв нет четких соответствий, но островные локальные соответствия между образами букв и кодами существуют, и именно в этом заключается польза содержательного кода. Это также касается случаев, когда нужно найти шрифт с количественно определёнными свойствами, например, при воспри-

ятии движущегося текста, или закреплённого текста с транспортного средства, движущегося с известными характеристическими скоростями; при восприятии текстов, которые приходится читать под углом, значительно отличающимся от прямого, написанных на неплоских поверхностях, наблюдаемых с помощью кривых зеркал при осмотре труднодоступных деталей конструкций и т.д. В связи с предыдущим можно упомянуть и проблему изучения семиотики движущихся надписей, которая оказалась одной из центральных в области изучения языкового ландшафта города[8] (ср. положение наблюдателя – *perspective point* – в том числе, нахождение на движущемся транспортном средстве – как одну из четырёх основных систем формирования образов – *principal imaging systems* – в когнитивной лингвистике [41, с. 254–255]).

Предложенная система для кодирования "H" прямого начертания при организации единого банка данных позволит:

1) находить одинаковые или похожие по начертанию каркасы;

2) целенаправленно получать монотонно меняющиеся серии каркасов, тем самым расширяя разнообразие шрифтов;

3) начать изучение психологии восприятия текста на количественном уровне в следующих направлениях:

а) вести статистику и выявлять зрительно-тематические связи – для каких видов текстов какие начертания употребляются более часто;

б) отслеживать тенденции и развитие моды на те или иные начертания;

в) выделять зоны наиболее удобочитаемых начертаний для прогнозирования создания новых шрифтов для определённых целей;

г) выявлять начертания, не употребляемые в дизайне вовсе, и попытаться выяснить причины этого.

Интегральность характеризаций шрифтов, использованная в *RHA*-кодах, конечно, требует некоторого навыка для успешной работы с ними, как и со всеми новыми инструментами.

Рассмотренный в настоящей статье метод может использоваться при кодировании географических карт, где ранговая формула строится, например, по соотношению площадей промежутков между горизонталями карты, по соотношению площадей разных ландшафтов на местности или геологических карт, для которых значимо соотношение площадей изображений пород – разного состава, возраста, разной степени оруденения. Возможен переход к кодированию объёмов или соотношений масс в объекте, например, гранулометрический анализ горных пород. Этот метод пригоден и для описания раскрасок камней [42], шкур животных, интерьеров, листовой мозаики деревьев, структуры сыпей в дерматологии и др.

Основная часть работы по методу *RHA* осуществлялась с использованием программы *Petros*3.2, составленной С.В. Мошкиным.

* * *

Авторы благодарят К.М. Кириченко за составление программы для анализа изображений букв; Н.А. Павлову, Т.М. Журавскую, К.А. Манукян за помощь в вопросах терминологии.

СПИСОК ЛИТЕРАТУРЫ

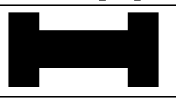
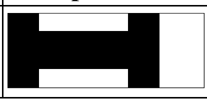
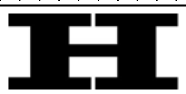
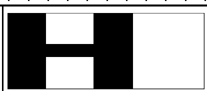
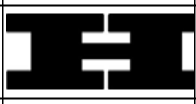
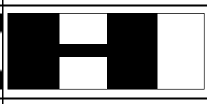
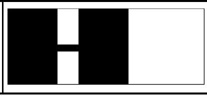
1. Сайт Fontproblem. – URL: http://www.promotion.su/link/25403 (дата обращения 21.12.2018)
2. Сайт ParaType. – URL: http://www.paratype.ru/ezine/issue01/synonims.htm
3. Королькова А. Живая типографика. – М.: IndexMarket, 2011. – *224 с.*
4. Брингхерст Р. Основы стиля в типографике. – М.: Издатель Д. Аронов, 2006. – 100 с.
5. Kapr A. Schriftkunst. – Dresden: VEB Verlag der Kunst, 1971. – 470 с.
6. YouWorkForThem раздел Fonts подраздел Serif . – URL: http://www.youworkforthem. com/fonts/serif (дата обращения 21.12.2018)
7. YouWorkForThem раздел Fonts подраздел Sans-serif. – URL: http://www.youworkforthem.com/fonts/sans-serif (дата обращения 21.12.2018)
8. YouWorkForThem раздел Fonts подраздел Stencil. – URL: http://www.youworkforthem. com/fonts/stencil (дата обращения 21.12.2018)
9. YouWorkForThem раздел Fonts подраздел Handwriting. – URL: http://www.youworkforthem.com/fonts/handwriting (дата обращения 21.12.2018)
10. MyFonts раздел Typographic category. – URL: http://www.myfonts.com/category/ (дата обращения 21.12.2018)
11. Каталог студии Letterhead 2001-2002 гг. – URL: http://issuu.com/letterhead/docs/lh_ font_catalog (дата обращения 21.12.2018))
12. Каталог студии Letterhead 1999 г. – URL: http://issuu.com/letterhead/docs/lhs_catalog_1999 (дата обращения 21.12.2018))
13. Фотонаборные шрифты / сост. Г. Козубов, В. Ефимов. – М.: Книга, 1983. – 176 с.
14. Цифровые шрифты. – М.: ПараТайп, 2004. – 508 с.
15. Monotype fonts by A-Z. – URL: http://catalog.monotype.com/alphabetical (дата обращения 10.10.2014)
16. Abstract Fonts раздел A-Z. – URL: http://www.abstractfonts.com/alpha/A (дата обращения 20.02.2019)
17. Font Spring раздел Alphabetical. – URL: http://www.fontspring.com/a2z?sort=alpha (дата обращения 20.02.2019)
18. Habr «Как найти нужный шрифт, не зная его названия?». – URL: https://habr.com/en/post/38046/ (дата обращения 20.02.2019).
19. My fonts by Monotype. – URL: http://www.myfonts.com/WhatTheFont/ (дата обращения 20.02.2019).
20. Identifont. – URL: http://www.identifont.com/identify.html (дата обращения 20.02.2019).
21. Шмелева А. Классификация шрифтов: практика и проблемы // Publish. – 2003. – № 1. – 80 с.
22. Moles A. Theorie de l'information et perception esthétique. – Paris: Flammarion, 1958; Моль А. Теория информации и эстетическое восприятие. – М.: Мир, 1966. – 352с.

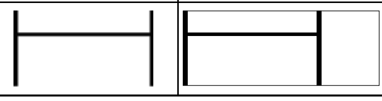
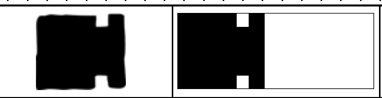
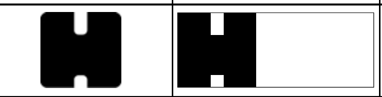
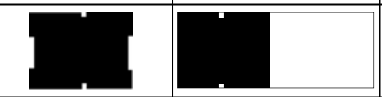
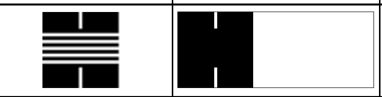
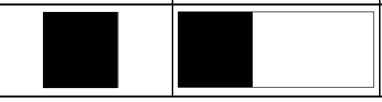
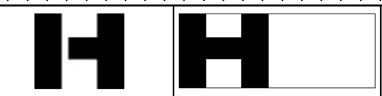
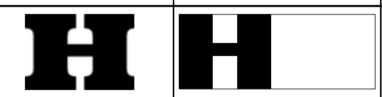
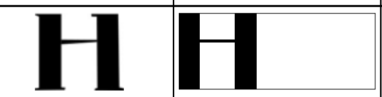
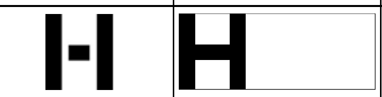
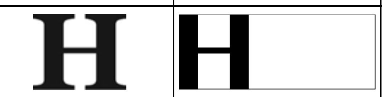
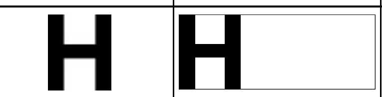
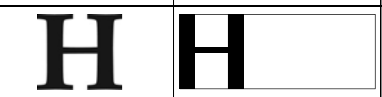
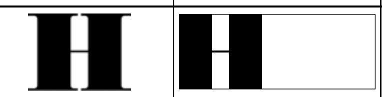
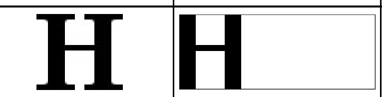
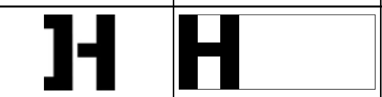
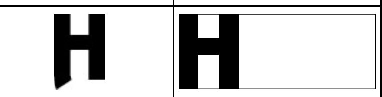
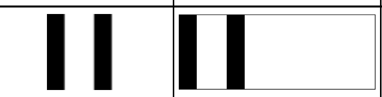
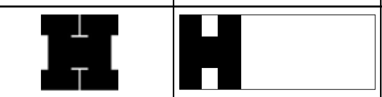
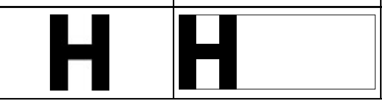
[8] См., например, серию статей в специальном выпуске журнала «International Journal Bilingualism». – 2014. – Вып. 18(5).

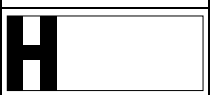
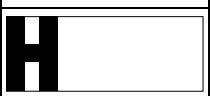
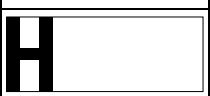
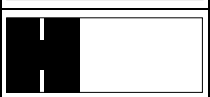
23. Петров Т.Г. Обоснование варианта общей классификации геохимических систем // Вестник ЛГУ. – 1971. – №18. – С. 30-38.
24. Петров Т.Г., Фарафонова О.И. Информационно-компонентный анализ. Метод RHA. – СПб, 2005. – 168 с.
25. Петров Т.Г. Информационный язык для описания составов многокомпонентных объектов // Научно-техническая информация. Сер 2. – 2001. – № 3. – С. 8-18; Petrov T.G. The RHA data language for describing the compositions of multicomponent systems // Automatic Documentation and Mathematical Linguistics. – 2001. – Vol. 35, № 2. – P. 8-22.
26. Чебанов С.В., Петров Т.Г. Интенсиональность, интенсиональные алфавиты, интенсиональные слова и словари // В сб. Актуальные проблемы современной когнитивной науки. – Иваново, 2013 – С. 239-266. DOI: 10.13140/RG.2.1.4542.8644 RG
27. Петров Т.Г. Метод RHA для кодирования, систематизации и отображения изменений возрастных составов населения. – URL: https://www.researchgate.net/profile/Tomas_Petrov/research2015 (дата обращения 20.02.19). DOI: 10.13140/RG.2.1.3207.2166.
28. Петров Т.Г. Графическое отображение процессов эволюции составов поликомпонентных объектов любой природы // Научно-техническая информация. Сер. 2. – 2012. – № 3. – С. 21–31; Petrov T.G. Graphic Representation of the Evolutionary Processes of the Compositions of Multicomponent Objects of Any Nature // Automatic Documentation and Mathematical Linguistics. – 2012. – Vol. 46, № 2. – P. 79-93. DOI: 10.3103/S0005105512020045
29. Petrov T.G., Moshkin S.V. RHA(T)-System for Coding of Discrete Distributions and Their Alteration Processes // Proc. The 3rd International Multi-Conference on Complexity, Informatics and Cybernetics IMCIC 2012. – Orlando, IIIS, 2012. – P. 12-16.
30. Петров Т.Г., Краснова Н.И. R-словарь-каталог химических составов минералов. – СПб: Наука, 2010. – 150 с.
31. Petrov T.G. “Soft” System of Coordinates in Regular Simplexes // International Journal of Intelligent Information Systems. – 2015. – Vol. 4, №1. – P. 1-7. DOI: 10.11648/ j.ijiis.20150401. 11/. – URL: http://article. sciencepublishinggroup.com/pdf/10. 11648.j.ijiis.20150401.11.pdf.
32. Петрова Е.Т., Петров Т.Г. Кодирование и систематизация шрифтов на базе информационного языка-метода RHA // Вестник СПГУ Технологии и дизайна. Сер 2. – 2015. – №1. – С. 39-44.
33. Седов Е.А. Одна формула и весь мир. Книга об энтропии. – М.: Знание, 1982. – 176 с.
34. Петров Т.Г. О мере сложности геохимических систем с позиций теории информации // Доклады АН СССР. – 1970 – Т.191, №4 – С.1094-1096.
35. Том Р. Структурная устойчивость и морфогенез. – М.: Логос, 2002.
36. Арнольд В.И., Афраймович В.С., Ильяшенко Ю.С., Шильников Л.П. Теория бифуркаций // Современные проблемы математики. Фундаментальные направления. Т. 5 (Итоги науки и техн. ВИНИТИ АН СССР). – М., 1985. – С. 5-218.
37. Арнольд В.И. Теория катастроф // Там же. – С. 219-277.
38. Думитраке Ю. Кенозис Христа в понимании Сергия Булгакова и Димитрия Стэнилоае // Дискуссия. – 2013. – №3(33). – С. 14-18.
39. Фуко М. Слова и вещи. Археология гуманитарных наук. – СПб: A-cad, 1994; Foucault M. Les mots et les choses une archeologie des sciences humaines – Paris: Gallimard, 1966.
40. Хомский Н. Синтаксические структуры // Новое в лингвистике. Вып. II. – М., 1962. – С. 412-527.
41. Talmy L. How language structures space // Spatial orientation: Theory, research, and application / eds. H.L. Pick, Jr., L.P. Acredolo. – NY; London, 1983. – P. 225–282.
42. Петров Т.Г., Шуйский А.В. Параметрическое описание рисунка цветного камня // Современные проблемы науки и образования – 2013. – № 5. – URL: https://science-education.ru/ru/article/view?id=10235.

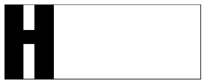
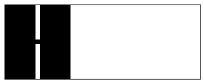
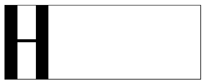
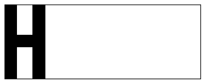
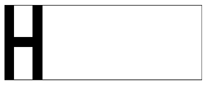
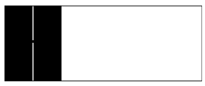
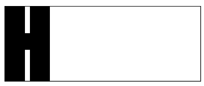
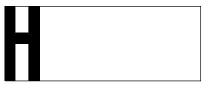
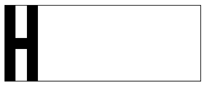
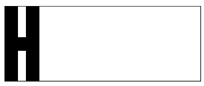
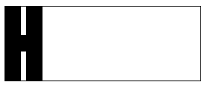
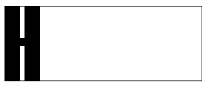
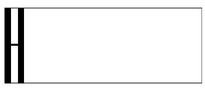
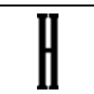
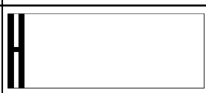
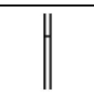
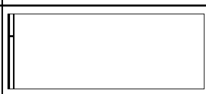
ПРИЛОЖЕНИЕ

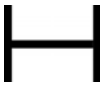
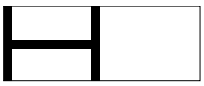
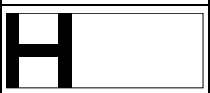
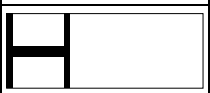
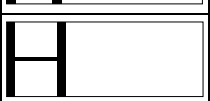
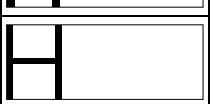
***RHA*- систематика каркасов "H"**

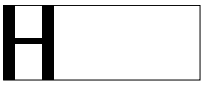
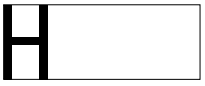
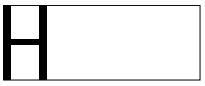
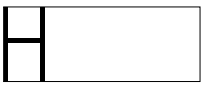
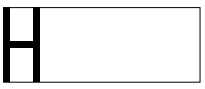
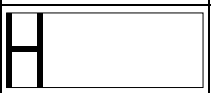
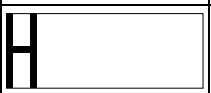
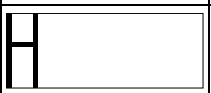
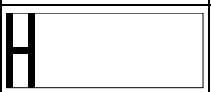
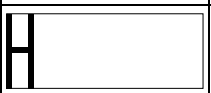
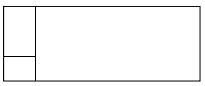
№	***R*-класс**	***H***	***A***	**Шрифт**	**Аллограф**	**Каркас в F**	**S_S%**	**B_S %**	**C_S %**	**O_S %**
1	S_s=B_s=C_s=O_s	1.386	0.000	*SuperCompl1			25.00	25.00	25.00	25.00
|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*										
2	S_s=$O_sC_sB_s$	1.187	0.303	King Tut Black			39.42	4.01	20.25	36.32
3	$S_sO_sC_sB_s$	1.134	0.340	Blackoak Std			52.01	4.01	20.24	23.74
4	$S_sO_sC_sB_s$	0.980	0.787	Guinness Extra Stout NF			50.7	0.88	9.73	38.7
|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*										

<table>
<tr><th>№</th><th>R-класс</th><th>H</th><th>A</th><th>Шрифт</th><th>Аллограф</th><th>Каркас в F</th><th>S_S%</th><th>B_S %</th><th>C_S %</th><th>O_S %</th></tr>
<tr><td>5</td><td>$C_sO_sS_sB_s$</td><td>0.900</td><td>0.678</td><td>Vienna Extended LET</td><td colspan="2">  </td><td>4.56</td><td>3.07</td><td>62.8</td><td>29.6</td></tr>
<tr><td colspan="11">|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*</td></tr>
<tr><td>6</td><td>$O_sS_sB_sC_s$</td><td>0.908</td><td>0.759</td><td>Cuadrifonte</td><td colspan="2">  </td><td>38.31</td><td>4.03</td><td>2.19</td><td>55.47</td></tr>
<tr><td>7</td><td>$O_sS_sB_s{=}C_s$</td><td>0.896</td><td>0.729</td><td>Tonal</td><td colspan="2">  </td><td>33.60</td><td>3.47</td><td>3.03</td><td>59.89</td></tr>
<tr><td>8</td><td>$O_sS_sB_sC_s$</td><td>0.796</td><td>1.373</td><td>YWFT Pudge</td><td colspan="2">  </td><td>44.77</td><td>2.06</td><td>0.33</td><td>52.85</td></tr>
<tr><td>9</td><td>$O_sS_sB_s{=}C_s$</td><td>0.747</td><td>1.378</td><td>Sinaloa</td><td colspan="2">  </td><td>36.87</td><td>0.85</td><td>0.82</td><td>61.46</td></tr>
<tr><td>10</td><td>$O_sS_s{=}B_sC_s$</td><td>0.665</td><td>+∞</td><td>Gropius Display</td><td colspan="2">  </td><td>38.20</td><td>0.00</td><td>0.00</td><td>61.80</td></tr>
<tr><td colspan="11">|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*</td></tr>
<tr><td>11</td><td>$O_sS_sC_sB_s$</td><td>1.103</td><td>0.341</td><td>Aksent</td><td colspan="2">  </td><td>27.47</td><td>5.22</td><td>12.80</td><td>54.51</td></tr>
<tr><td>12</td><td>$O_sS_sC_sB_s$</td><td>1.020</td><td>0.571</td><td>Egyptienne Extd D Bold</td><td colspan="2">  </td><td>34.19</td><td>2.09</td><td>10.47</td><td>53.26</td></tr>
<tr><td>13</td><td>$O_sS_sC_sB_s$</td><td>0.970</td><td>0.808</td><td>Quadrata</td><td colspan="2">  </td><td>21.63</td><td>0.69</td><td>17.00</td><td>60.68</td></tr>
<tr><td>14</td><td>$O_sS_sC_sB_s$</td><td>0.966</td><td>0.491</td><td>Glaser Stencil</td><td colspan="2">  </td><td>16.34</td><td>3.63</td><td>14.00</td><td>66.03</td></tr>
<tr><td>15</td><td>$O_sS_sC_sB_s$</td><td>0.937</td><td>0.713</td><td>Georgia</td><td colspan="2">  </td><td>19.16</td><td>1.21</td><td>15.05</td><td>64.58</td></tr>
<tr><td>16</td><td>$O_sS_sC_sB_s$</td><td>0.911</td><td>0.582</td><td>PT_PTC75 Caption Bold</td><td colspan="2">  </td><td>16.31</td><td>2.76</td><td>12.31</td><td>68.62</td></tr>
<tr><td>17</td><td>OS=CB</td><td>0.909</td><td>0.704</td><td>Book Antiqua</td><td colspan="2">  </td><td>15.86</td><td>1.41</td><td>15.59</td><td>67.14</td></tr>
<tr><td>18</td><td>$O_sS_sC_sB_s$</td><td>0.908</td><td>1.169</td><td>Falstaff</td><td colspan="2">  </td><td>33.66</td><td>0.22</td><td>8.54</td><td>57.58</td></tr>
<tr><td>19</td><td>$O_sS_sC_sB_s$</td><td>0.884</td><td>0.773</td><td>SchoolBook_Bold</td><td colspan="2">  </td><td>16.74</td><td>1.12</td><td>13.88</td><td>68.25</td></tr>
<tr><td>20</td><td>$O_sS_sC_sB_s$</td><td>0.878</td><td>0.660</td><td>Project Fairfax</td><td colspan="2">  </td><td>18.84</td><td>2.21</td><td>9.65</td><td>69.31</td></tr>
<tr><td>21</td><td>$O_sS_sC_sB_s$</td><td>0.866</td><td>0.667</td><td>Regata</td><td colspan="2">  </td><td>19.61</td><td>2.40</td><td>8.25</td><td>69.74</td></tr>
<tr><td>22</td><td>$O_sS_sC_sB_s$</td><td>0.866</td><td>+∞</td><td>GHSans</td><td colspan="2">  </td><td>18.20</td><td>0.00</td><td>15.16</td><td>66.65</td></tr>
<tr><td>23</td><td>$O_sS_sC_sB_s$</td><td>0.864</td><td>0.669</td><td>Thickset</td><td colspan="2">  </td><td>23.33</td><td>3.42</td><td>4.94</td><td>68.32</td></tr>
<tr><td>24</td><td>$O_sS_sC_sB_s$</td><td>0.861</td><td>0.677</td><td>PT_PTS85 Extra Bold</td><td colspan="2">  </td><td>17.75</td><td>2.31</td><td>9.35</td><td>70.58</td></tr>
</table>

№	*R*-класс	*H*	*A*	Шрифт	Аллограф	Каркас в F	S_S%	B_S %	C_S %	O_S %
25	$O_sS_sC_sB_s$	0.860	0.664	PT_PTS75 Bold			15.07	2.24	11.43	71.27
26	$O_sS_sC_sB_s$	0.858	0.671	PT_PTS65 Demi			15.06	2.17	11.46	71.31
27	$O_sS_sC_sB_s$	0.857	0.993	BodoniBold			18.16	0.46	12.85	68.53
28	$O_sS_sC_sB_s$	0.856	0.736	Yess_Bold			24.35	2.15	5.81	67.69
29	$O_sS_sC_sB_s$	0.848	0.713	PT_PTS95 Black			20.67	2.18	7.16	69.99
30	$O_sS_sC_sB_s$	0.839	0.677	Century Gothic			14.80	2.35	10.34	72.51
31	$O_sS_sC_sB_s$	0.825	0.845	YWFT Black Slabbath			27.25	1.80	4.05	66.89
32	$O_sS_sC_sB_s$	0.825	1.294	Braggadocio			42.77	0.30	3.20	53.73
33	$O_sS_sC_sB_s$	0.822	0.935	Acsioma_Shock			31.78	1.50	3.06	63.65
34	$O_sS_sC_sB_s$	0.810	0.734	FF Archian Stencil Pro			16.54	2.13	8.04	73.29
35	$O_sS_sC_sB_s$	0.809	0.947	Acsioma_Next Rough			30.52	1.53	2.91	65.05
36	$O_sS_sC_sB_s$	0.806	0.975	Acsioma_Medium			31.47	1.44	2.71	64.38
37	$O_sS_sC_sB_s$	0.800	1.007	Acsioma_Super Shock			31.75	1.29	2.64	64.33
38	$O_sS_sC_sB_s$	0.797	0.980	SchoolBook_Cond Bold			31.78	1.33	2.50	64.38
39	$O_sS_sC_sB_s$	0.797	1.012	Acsioma_Next			16.94	0.63	10.04	72.39
40	$O_sS_sC_sB_s$	0.777	0.793	PT_PTN87 Nar-row Extra Bold			15.92	1.84	7.48	74.76
41	$O_sS_sC_sB_s$	0.758	0.876	PT_PTN97 Narrow Black			19.07	1.59	5.22	74.12
42	$O_sS_sC_sB_s$	0.756	1.008	Adamant			23.24	1.01	4.13	71.62
43	$O_sS_sC_sB_s$	0.753	0.828	PT_PTN77 Narrow Bold			13.57	1.63	8.44	76.36
44	$O_sS_sC_sB_s$	0.752	1.289	Avatar			34.82	0.80	1.28	63.10

№	*R*-класс	*H*	*A*	Шрифт	Аллограф	Каркас в F	S_S%	B_S %	C_S %	O_S %
45	$O_sS_sC_sB_s$	0.726	1.558	Fatta			36.97	0.46	0.73	61.84
46	$O_sS_sC_sB_s$	0.724	0.991	Diamonds			19.03	1.10	4.71	75.16
47	$O_sS_sC_sB_s$	0.723	1.657	Gaslon			31.29	0.12	2.07	66.52
48	$O_sS_sC_sB_s$	0.694	0.970	PT_PTS87 Cond Extra Bold			15.10	1.22	5.62	78.07
49	$O_sS_sC_sB_s$	0.693	1.227	BodoniCondC			12.69	0.32	9.11	77.88
50	$O_sS_sC_sB_s$	0.691	0.935	PT_PTS77 Cond Bold			12.34	1.31	7.26	79.10
51	$O_sS_sC_sB_s$	0.689	1.249	Garbage			17.65	0.36	5.42	76.56
52	$O_sS_sC_sB_s$	0.675	1.074	PT_PTS97 Cond Black			17.76	1.03	3.75	77.46
53	$O_sS_sC_sB_s$	0.657	0.983	PT_PTS67 Cond Demi			9.68	1.18	8.30	80.84
54	$O_sS_sC_sB_s$	0.642	2.333	Loudine			27.98	0.02	0.87	71.12
55	$O_sS_sC_sB_s$	0.640	1.291	Impact			19.94	0.64	2.27	77.14
56	$O_sS_sC_sB_s$	0.614	1.082	Hill			11.08	0.99	5.72	82.21
57	$O_sS_sC_sB_s$	0.591	1.144	PT_PTS79 Extra Cond Bold			10.91	0.85	5.22	83.02
58	$O_sS_sC_sB_s$	0.582	1.235	PT_PTS89 Extra Cond Extra Bold			13.51	0.74	3.40	82.35
59	$O_sS_sC_sB_s$	0.581	1.352	PT_PTS99 Extra Cond Black			16.26	0.61	2.18	80.96
60	$O_sS_sC_sB_s$	0.570	1.165	PT_PTS69 Extra Cond Demi			8.31	0.79	6.68	84.22
61	$O_sS_sC_sB_s$	0.544	1.585	CompactBold			15.33	0.26	2.10	82.31
62	$O_sS_sC_sB_s$	0.389	1.895	Radar			5.82	0.10	3.79	90.29
63	$O_sS_sC_sB_s$	0.344	1.938	Titanic Condensed			6.22	0.14	2.11	91.52
64	$O_sS_sC_sB_s$	0.177	2.613	Super C			1.96	0.04	1.49	96.51
|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*										

№	*R*-класс	*H*	*A*	Шрифт	Аллограф	Каркас в F	S_S%	B_S %	C_S %	O_S %
65	$O_sC_sS_sB_s$	1.118	0.382	Flat10 ArtDeco			19.20	3.34	25.40	52.05
66	$O_sC_sS_sB_s$	1.047	0.451	Distill			7.84	4.39	36.50	51.27
67	$O_sC_sS_sB_s$	0.998	0.476	Ecyr			10.46	3.77	23.83	61.94
68	$O_sC_sS_sB_s$	0.955	0.713	Ustav			16.08	1.13	19.70	63.09
69	$O_sC_sS_sB_s$	0.947	0.520	YWFT LED			13.26	3.34	16.45	66.94
70	$O_sC_sS_sB_s$	0.903	0.600	Yess_Regular			11.86	2.61	16.64	68.89
71	$O_sC_sS_sB_s$	0.899	0.593	Lodge			6.40	3.79	22.26	67.54
72	$O_sC_sS_sB_s$	0.893	0.618	PT_PTC55 Caption Regular			11.04	2.50	17.25	69.20
73	$O_sC_sS_sB_s$	0.874	0.651	Arial			10.09	2.33	17.56	70.02
74	$O_sC_sS_sB_s$	0.864	0.661	Verdana			10.16	2.30	16.82	70.72
75	$O_sC_sS_sB_s$	0.863	0.768	School-Book_Book			12.45	1.26	16.51	69.77
76	$O_sC_sS_sB_s$	0.862	0.765	Bengaly			12.82	1.28	15.97	69.93
77	$O_sC_sS_sB_s$	0.848	0.908	Bodoni			13.17	0.69	16.32	69.82
78	$O_sC_sS_sB_s$	0.837	0.702	Tahoma			10.18	2.07	15.49	72.26
79	$O_sC_sS_sB_s$	0.835	0.704	Kabel			9.28	2.12	16.50	72.10
80	$O_sC_sS_sB_s$	0.826	0.898	Pacioli			8.20	0.89	21.09	69.82
81	$O_sC_sS_sB_s$	0.824	0.864	Times New Roman			10.81	0.95	16.77	71.46
82	$O_sC_sS_sB_s$	0.822	0.732	PT_PTS55 Regular			9.34	1.92	16.01	72.73
83	$O_sC_sS_sB_s$	0.814	0.766	PT_PTS45 Light			8.31	1.74	17.36	72.59
84	$O_sC_sS_sB_s$	0.795	0.841	Calipso			6.16	1.50	20.29	72.05

№	*R*-класс	*H*	*A*	Шрифт	Аллограф	Каркас в F	S_S%	B_S %	C_S %	O_S %
85	$O_sC_sS_sB_s$	0.781	0.876	School-Book_Cond			11.33	1.06	13.15	74.46
86	$O_sC_s=S_sB_s$	0.748	0.825	PT_PTN67 Narrow Demi			10.61	1.62	10.88	76.89
87	$O_sC_sS_sB_s$	0.726	0.883	PT_PTN57 Narrow Regular			7.97	1.41	13.11	77.51
88	$O_sC_sS_sB_s$	0.703	0.950	PT_PTN47 Narrow Light			7.39	1.14	13.24	78.23
89	$O_sC_sS_sB_s$	0.669	0.999	Moon Star Soul			8.76	1.01	10.12	80.11
90	$O_sC_sS_sB_s$	0.650	1.118	Jatran			4.28	0.84	15.68	79.21
91	$O_sC_sS_sB_s$	0.635	1.035	PT_PTS57 Cond Regular			6.80	1.06	10.59	81.56
92	$O_sC_sS_sB_s$	0.625	1.059	Circus			7.16	0.98	9.83	82.03
93	$O_sC_sS_sB_s$	0.624	1.097	PT_PTS47 Cond Light			5.47	0.90	12.11	81.52
94	$O_sC_sS_sB_s$	0.560	1.176	Zirkus			6.55	0.80	7.99	84.65
95	$O_sC_sS_sB_s$	0.551	1.249	Rustica			4.65	0.66	10.20	84.49
96	$O_sC_s=S_sB_s$	0.533	1.240	PT_PTS59 Extra Cond Regular			6.75	0.69	6.87	85.69
97	$O_sC_sS_sB_s$	0.512	1.304	PT_PTS49 Extra Cond Light			5.47	0.59	7.61	86.34
98	$O_sC_sS_sB_s$	0.473	2.083	FOSU			0.61	0.12	15.27	84.00
99	$O_sC_sS_sB_s$	0.263	2.185	FR Pasta Mono			1.99	0.09	3.71	94.22

Материал поступил в редакцию 27.12.18.

Сведения об авторах

ПЕТРОВА Екатерина Томасовна – художник-график, независимый исследователь, Санкт-Петербург
e-mail: katia.petrova@gmail.com

ПЕТРОВ Томас Георгиевич – доктор геолого-минералогических наук, профессор, консультант фирмы ООО "Соколов" Санкт-Петербург
e-mail: tomas_petrov@rambler.ru

ЧЕБАНОВ Сергей Викторович – доктор филологических наук, профессор кафедры математической лингвистики Санкт-Петербургского государственного университета
e-mail: s.chebanov@gmail.com

МОШКИН Сергей Владимирович – кандидат геолого-минералогических наук, ведущий программист Российской национальной библиотеки, Санкт-Петербург
e-mail: svmoshkin52@gmail.com