ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с сентября 2010 г. DOI 10.17587/issn.2220-3397 ISSN 2220-3397

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

СОДЕРЖАНИЕ

Журнал зарегистрирован в Федеральной службе по надзору в сфере связи, информационных технологий и массовых коммуникаций. Свидетельство о регистрации ПИ № ФС77-38590 от 24 декабря 2009 г.

Журнал распространяется по подписке, которую можно оформить в любом почтовом отделении (индексы: по каталогу агентства "Роспечать" — **22765**, по Объединенному каталогу "Пресса России" — **39795**) или непосредственно в редакции.
Тел.: (499) 269-53-97. Факс: (499) 269-55-10.
Http://novtex.ru/prin/rus E-mail: prin@novtex.ru
Журнал включен в систему Российского индекса научного цитирования.
Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

THEORETICAL AND APPLIED SCIENTIFIC AND TECHNICAL JOURNAL

SOFTWARE ENGINEERING

PROGRAMMNAYA INGENERIA

Vol. 7
N 9
2016

Published since September 2010 **DOI 10.17587/issn.2220-3397** **ISSN 2220-3397**

CONTENTS

Information about the jornal is available online at:
http://novtex.ru/prin/eng e-mail:prin@novtex.ru

УДК 004.222 DOI: 10.17587/prin.7.387-394

К. С. Исупов, канд. техн. наук, вед. науч. сотр., e-mail: ks_isupov@vyatsu.ru,
В. С. Князьков, д-р техн. наук, проф., e-mail: knyazkov@vyatsu.ru,
А. С. Куваев, магистрант, e-mail: kyvaevy@gmail.com,
М. В. Попов, магистрант, e-mail: mihailpopov1993@mail.ru,
Вятский государственный университет, г. Киров

Разработка пакета высокоточной арифметики для суперкомпьютеров с графическими ускорителями

Представлена программная библиотека арифметики многократной точности для параллельных вычислительных систем с гибридной CPU/GPU-архитектурой. Предусмотрена поддержка трех числовых типов данных: числа фиксированной точности с расширенной экспонентой, целые числа произвольной длины и числа с плавающей точкой произвольной длины. Особенность библиотеки — использование системы остаточных классов для представления многоразрядных мантисс, что допускает их эффективную параллельную обработку в силу отсутствия переносов между цифрами обрабатываемых чисел. Проблемные для системы остаточных классов операции, такие как сравнение и контроль переполнения, реализуются с использованием интервального метода оценки относительных (дробных) значений чисел.

Ключевые слова: *вычисления многократной точности, расширенный диапазон, система остаточных классов, CUDA, программное обеспечение*

Введение

Научные вычисления часто требуют поддержки числовых типов данных, разрядность которых превышает стандартный формат двойной точности (binary64). Потребность в таких вычислениях возникает, в частности, при изучении динамических систем [1], в процессе расчета моментов произвольных порядков в хроматографии [2], при численном исследовании стабильности объектов Солнечной системы [3], в задачах моделирования климата [4] и т. д. Ввиду большого объема вычислений, производительность считается одним из наиболее важных нефункциональных требований в таких приложениях. Это приводит к необходимости использования современных параллельных вычислительных систем, реализованных на базе многоядерных процессоров.

Поскольку многоразрядные типы данных изначально не поддерживаются в аппаратном обеспечении общего назначения, используется арифметика многократной точности — программная эмуляция вычислений с числами большой разрядности. Встроенная поддержка такой эмуляции реализована лишь в некоторых языках программирования, поэтому более общим вариантом является использование высокоточных библиотек. К их числу относятся такие библиотеки, как The GNU Multiple Precision Arithmetic Library (GMP) [5], The GNU MPFR Library [6], Library of Efficient Data Types and Algorithms (LEDA) [7], ARPREC [8]. Перечисленные и аналогичные библиотеки ориентированы на универсальные процессорные платформы общего назначения (CPU). Вместе с тем современные графические ускорители (GPU) являются мощным и экономически эффективным средством для научных вычислений [9]. Их многоядерные процессоры обеспечивают высокую степень параллелизма при использовании современных технологий многопоточной обработки данных. В связи с этим обстоятельством вычислительные платформы многих современных кластерных систем обладают гибридной архитектурой — в качестве средств обработки данных используются как CPU, так и GPU.

Поскольку технология применения графических процессоров видеокарт для вычислений общего назначения (GPGPU) появилась сравнительно недавно, в настоящее время существует ограниченное число библиотек многократной точности для GPU. В 2010 г. K. Zhao и X. Chu разработали программный пакет GPUMP [10]. Этот пакет не поддерживает арифметику с плавающей точкой. В 2011 г. T. Nakayama и D. Takahashi представили библиотеку с плавающей точкой CUMP [11], которая представляет собой вариант переноса на GPU библиотеки GMP. К сожалению, в настоящее время функциональные возможности CUMP предельно ограничены — поддерживаются лишь три базовые арифметические операции: сложение, вычитание и умножение. Из высокоточных GPU-библиотек, реализующих все арифметиче-

ские операции, тригонометрию и ряд других математических функций, следует отметить GARPREC и GQD [12], причем последняя поддерживает лишь расширенную точность (форматы "double-double" и "quad-double"). Кроме всего прочего, перечисленные библиотеки ориентированы исключительно на GPU, а для поддержки высокоточной арифметики на CPU необходимо использовать другие средства. Это усложняет разработку гибридных параллельных программ и дополнительно требует использования специализированных алгоритмов преобразования между различными высокоточными типами данных.

Необходимо отметить, что типы данных, реализованные в известных высокоточных библиотеках (как для CPU, так и для GPU), основаны на традиционных последовательных методах длинной арифметики и реализуют лишь последовательную обработку одного многоразрядного числа. Учитывая высокую сложность алгоритмов, это негативно сказывается на скорости вычислений и, соответственно, на времени решения прикладных задач.

Принимая во внимание изложенные выше причины, важной задачей является создание эффективных алгоритмов и программного обеспечения для поддержки вычислений многократной точности на высокопроизводительных системах с гибридной CPU/GPU-архитектурой. В работе рассмотрен один из вариантов решения указанной задачи, основанный на представлении чисел большой разрядности в виде разложений по модулям системы остаточных классов (СОК). Такой подход позволяет организовать параллельное (до уровня отдельных цифр мантисс) выполнение высокоточных операций, эффективно задействовав имеющиеся ресурсы современных вычислительных архитектур.

Особенности вычислений с многократной точностью при использовании систем счисления в остаточных классах

Основной недостаток позиционных систем счисления при реализации вычислений повышенной точности связан с образованием длинных цепочек переносов между разрядами чисел, что приводит к невозможности (в явной форме) распараллеливания высокоточных алгоритмов и существенному возрастанию времени обработки. Указанный недостаток обуславливает неэффективное использование как SIMD-расширений современных CPU, так и альтернативных параллельных платформ, в частности, GPU. В этом отношении привлекательными выглядят непозиционные способы кодирования информации, позволяющие разделить трудоемкие операции по обработке многоразрядных чисел на несколько операций меньшей разрядности, выполняемых параллельно.

Система остаточных классов [13—15] — непозиционная система счисления, являющаяся альтернативой двоичному представлению чисел. Эта система задается набором попарно взаимно простых модулей $\{m_1, m_2, \ldots, m_n\}$. Динамический диапазон определяется произведением $M = m_1 \times m_2 \times \cdots \times m_n$. Любое целое X от 0 до $M - 1$ в СОК однозначно представляется n-кортежем наименьших неотрицательных вычетов $\langle x_1, x_2, \ldots, x_n \rangle$, где $x_i = |X|_{m_i}$, что равносильно решению сравнения $x_i \equiv X \bmod m_i$. Остатки x_i не зависят друг от друга, поэтому такие многоразрядные арифметические операции, как сложение, вычитание и умножение могут осуществляться без какого-либо распространения переноса (между остатками), что в большинстве случаев является основным ограничивающим фактором в позиционных системах счисления. Благодаря этому СОК широко используется во многих областях высокоскоростной компьютерной арифметики, в частности, в цифровой обработке сигналов [16], криптографии [17], для обнаружения и исправления ошибок кодирования [18], в цифровой обработке изображений [19].

Недостатком СОК является сложность операций, требующих оценки значения числа, таких как сравнение, определение знака, обнаружение переполнения. Классический способ выполнения этих операций основан на китайской теореме об остатках [14] и состоит в вычислении двоичных представлений чисел с их последующим анализом. Такой способ является трудоемким, так как требует выполнения многоразрядных умножений и сложений с последующей редукцией по большому модулю M. Многие другие методы оценки значения в СОК базируются на преобразовании в систему со смешанными основаниями (*Mixed-Radix Conversion*) [13, 14]. Однако на практике они оказываются неэффективными, так как требуют большого объема вычислений и/или использования больших подстановочных таблиц.

Альтернативный метод [20] основан на вычислении и анализе интервальной функции от остатков, локализующей относительное значение числа в СОК — интервально-позиционной характеристике (ИПХ) $I(X/M)$, которая представляется двумя направленно округленными числами — $\underline{X/M}$ и верхней границей $\overline{X/M}$ — и локализует в единичном полуинтервале значение X, масштабированное относительно произведения модулей M так, что $\underline{X/M} \leqslant X/M \leqslant \overline{X/M}$. Границы ИПХ — числа с плавающей точкой машинной точности, вычисляемые с использованием направленных округлений. Вычисление границ ИПХ на основе остатков числа проводится за линейное и логарифмическое время в последовательном и параллельном случаях соответственно. Кроме этого, над ИПХ определены арифметические операции. Сложение и умножение ИПХ выполняется в соответствии с правилами, справедливыми для обычных вещественных интервалов. Умножение и деление определяются с учетом относительности выражаемых величин: дополнительно учитывается вес $1/M$. Основные аспекты использования ИПХ для выполнения проблемных операций в СОК рассмотрены в работе [20].

Пакет высокоточных вычислений для систем с гибридной архитектурой

Авторами разрабатывается пакет **MPRES** — **M**ultiple-**P**recision **Res**idue-based Arithmetic Library — библиотека языков C/C++ для вычислений многократной точности на системах с гибридными узлами. В библиотеке используются новые типы данных произвольной длины, допускающие параллельную обработку. К настоящему времени предусмотрена поддержка центральных процессоров и графических ускорителей. Часть библиотеки для GPU ориентирована на архитектуру параллельных вычислений CUDA. Прототипом является разработанная ранее библиотека MF-Library для CPU, которая при решении ряда задач оказалась эффективнее лучших мировых аналогов [21].

Форматы и структуры данных

В пакете MPRES реализуется поддержка трех классов чисел: числа машинной точности с расширенной экспонентой (*extended-range floating-point*), целые числа произвольной точности (*modular integer*) и числа с плавающей точкой произвольной точности (*modular-positional floating-point*). Процедуры обработки этих классов чисел составляют арифметический уровень пакета (рис. 1). Далее рассмотрим подробнее каждый из форматов данных.

Extended-Range Floating-Point. Формат с расширенной экспонентой (ER-формат) не увеличивает вычислительную точность, но участвует в образовании двух других форматов многократной точности. Кроме этого, поддержка арифметики расширенного диапазона позволяет минимизировать вероятность переполнения (*Overflow Exception*) или потери значимости (*Underflow Exception*) в процессе вычислений и может быть востребована в приложениях, связанных с обработкой экстремально больших или маленьких по абсолютному значению величин.

Формат с расширенной экспонентой строится посредством объединения машинного целого i со стандартным машинным числом с плавающей точкой f. Пара (f, i) рассматривается как число $x = f \times B^i$, где B — основание экспоненты (константа) [22]. Предполагается, что основание B однозначно определено в контексте. По умолчанию f — стандартное число двойной точности; i — 32-битное целое со знаком ($-2\ 147\ 483\ 648 \leqslant i \leqslant 2\ 147\ 483\ 647$), $B = 2^{256}$, что обеспечивает диапазон представимых чисел $>10^{\pm 165\ 492\ 990\ 270}$.

В пакете MPRES числа с расширенной экспонентой представлены пользовательскими типами данных `er _ t`, `er _ arr _ t` и `er _ static _ t` (рис. 2). На рис. 3 приведены алгоритмы "регулировки", сложения и умножения чисел с расширенной экспонентой. Функция регулировки `er _ adjust` выполняет нормализацию числа, под которой здесь понимается расположение мантиссы `frac` в интервале ($1/B$; B). При сложении вызывается функция циклической регулировки `er _ badjust`, поскольку однократного выполнения `er _ adjust` может оказаться недостаточно при сложении близких по значению чи-

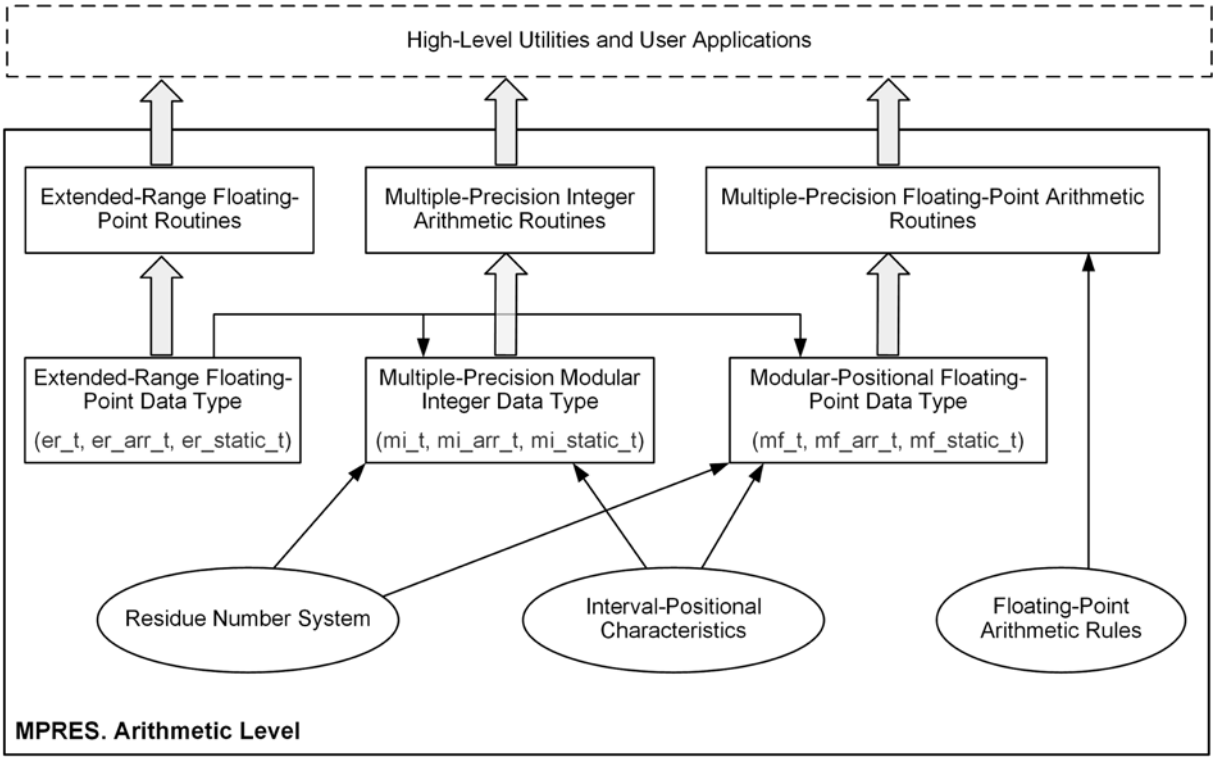


Рис. 1. Типы данных MPRES

```
/*Internal structure*/
typedef struct {
    er_frac_t frac;   //Significand
    er_exp_t  exp;    //Exponent
} __extended_range_struct;

/*User data types*/
typedef __extended_range_struct *er_t;          //For single number
typedef __extended_range_struct *er_arr_t;      //For arrays
typedef __extended_range_struct  er_static_t;   //For device side code
```

Рис. 2. Типы данных с расширенной экспонентой: `er _ frac _ t` соответствует стандартному числу с плавающей точкой одинарной (float) или двойной (double) точности; `er _ exp _ t` — целое число со знаком

```
er_t er_adjust(er_t x) {
    if (x.frac = 0) then
        return 0
    er_t z := x
    if (abs(x.frac) ≥ B) then
        z.frac := z.frac / 2^Log(B)
        z.exp  := z.exp  + 1
    else if (abs(x.frac) ≤ 1/B) then
        z.frac := z.frac × 2^Log(B)
        z.exp  := z.exp  - 1
    end if
    return z
}
```

```
er_t er_add(er_t x, er_t y) {
    if (x.frac = 0 and x.exp = 0) then
        return y
    else if (y.frac = 0 and y.exp = 0) then
        return x
    er_t z
    er_exp_t d := abs(x.exp - y.exp)
    if (x.exp > y.exp) then
        z.frac := x.frac + y.frac × 2^-d × Log(B)
        z.exp  := x.exp
    else if (y.exp > x.exp) then
        z.frac := y.frac + x.frac × 2^-d × Log(B)
        z.exp  := y.exp
    else
        z.exp  := x.exp
        z.frac := x.frac + y.frac
    end if
    return er_badjust(z)
}
```

```
er_t er_badjust(er_t x) {
  er_t z := er_adjust(x)
  while(x.exp≠z.exp or x.frac≠z.frac) do
      x := z;
      z := er_adjust(x)
  end while
  return z
}
```

```
er_t er_mul(er_t x, er_t y) {
    er_t z
    z.exp  := x.exp + y.exp
    z.frac := x.frac × y.frac
    return er_adjust(prod)
}
```

Рис. 3. Сложение и умножение представлений с расширенной экспонентой

сел разных знаков. Предполагается, что основание экспоненты B — степень двойки, при этом $\mathrm{Log}(B)$ обозначает показатель этой степени, т. е. двоичный логарифм от B.

Multiple-Precision Modular Integer. В данном формате целое число произвольной длины представляется знаком s, мантиссой в СОК X, а также ИПХ мантиссы $I(X/M)$. Мантисса X выражает абсолютное значение числа и представлена кортежем остатков $x_1, x_2, \ldots, x_n$ по модулям $m_1, m_2, \ldots, m_n$. Мантисса интерпретируется как целое число в интервале $[0, M-1]$, где M — динамический диапазон СОК. ИПХ локализует в единичном интервале значение X, масштабированное относительно M. Границы ИПХ $\underline{X/M}$ и $\overline{X/M}$ представлены в виде чисел с расширенной экспонентой (тип `er _ t`), что позволяет использовать ИПХ без ограничений на M.

Для компактной записи данного числового представления, названного MI-формат, используется нотация $x \to \{s, X, I(X/M)\}$, которая выражает значение

$$x = (-1^s) \times \left| \sum_{i=1}^{n} x_i \mid M_i^{-1} \mid_{m_i} M_i \right|_M,$$

где $M_i = M/m_i$, $\mid M_i^{-1} \mid_{m_i}$ — мультипликативная инверсия M_i по отношению к m_i.

Если $x \geqslant 0$, то $s = 0$; если $x < 0$, то $s = 1$. Явное выделение знака, которое стало эффективным благодаря введению ИПХ в представление числа, делает процесс вычислений более регулярным и позволяет избавиться от разбиения модулярного диапазона на области положительных и отрицательных чисел.

Операции над числами, представленными в многоразрядном MI-формате, выполняются достаточно просто. К примеру, произведением чисел x и y является число $z \to \{s_z, Z, I(Z/M)\}$, где $s_z = s_x \oplus s_y$, $Z = |X \times Y|_M$, $I(Z/M) = I(X/M) \times I(Y/M)$. При этом, если $\underline{Z/M} > \frac{M-1}{M}$, то при умножении мантисс возникло переполнение. И напротив, если $\overline{Z/M} \leqslant \frac{M-1}{M}$, то переполнения не возникло. Все цифры мантисс вычисляются параллельно:

$$Z = |X \times Y|_M = \begin{cases} z_1 = x_1 \times y_1 \mod m_1, \\ z_2 = x_2 \times y_2 \mod m_2, \\ \ldots \quad \ldots \quad \ldots \quad \ldots \\ z_n = x_n \times y_n \mod m_n. \end{cases}$$

Для нахождения разности $z = x - y$, когда $s_x = s_y$, выполняется сравнение $I(X/M)$ и $I(Y/M)$: если $\underline{X/M} > \overline{Y/M}$, то $s_z = s_x$, $Z = |X - Y|_M$, $I(Z/M) = I(X/M) - I(Y/M)$. Если же $\overline{X/M} < \underline{Y/M}$, то $s_z = \neg s_x$, $Z = |Y - X|_M$, $I(Z/M) = I(Y/M) - I(X/M)$. Если ИПХ мантисс пересекаются, то знак результата требует уточнения.

Разрядность числа (в битах), представленного описанным способом, определяется значением $\lfloor \log_2 M \rfloor$. Алгоритм генерации модулей СОК, обеспечивающих заданную точность (разрядность) вычислений, состоит из следующих шагов.

Шаг 1. Для заданных точности ε и числа модулей n определяется нечетное целое неотрицательное число m_1, удовлетворяющее неравенству $\lfloor \log_2 m_1 \rfloor \geqslant \varepsilon/n$.

Шаг 2. Методом перебора, начиная с $m_1 + 2$, выбирается второй модуль m_2, как наименьшее целое нечетное число, превышающее m_1 и попарно взаимно простое с ним. Проверка взаимной простоты чисел выполняется поиском наибольшего общего делителя по алгоритму Евклида [23].

Шаг 3. Выбирается модуль m_3 — наименьшее нечетное целое неотрицательное число, превышающее m_2 и попарно взаимно простое с m_1 и m_2.

Шаг 4. Аналогичным образом выбираются оставшиеся модули m_i, $4 \leqslant i \leqslant n$. При этом каждый i-й модуль должен удовлетворять условию попарной взаимной простоты с остальными.

```
/*Internal structure*/
typedef struct {
     mi_sig_t s;                      //Sign
     mi_res_t residue[MODULI_SIZE];   //Significand in RNS
     er_t     ipc_low;                //Lower IPC bound
     er_t     ipc_upp;                //Upper IPC bound
} __mp_integer_struct;

/*User data types*/
typedef __mp_integer_struct *mi_t;          //For single number
typedef __mp_integer_struct *mi_arr_t;      //For arrays
typedef __mp_integer_struct  mi_static_t;   //For device side code
```

Рис. 4. Целочисленные модулярные типы данных произвольной длины: `mi _ sig _ t` — короткое целое число; `mi _ res _ t` — машинное целое со знаком

В результате работы алгоритма будет получен набор $\{m_1, m_2, \ldots, m_n\}$ такой, что $\log_2(m_1 \times m_2 \times \ldots \times m_n) \geqslant \varepsilon$. Выполнение представленного алгоритма проводится однократно при инициализации библиотеки в случае, если разрядность вычислений необходимо изменить.

MI-формат реализуется в MPRES типами данных `mi _ t`, `mi _ arr _ t` и `mi _ static _ t` (рис. 4). Схема многоразрядных вычислений в MI-формате на GPU представлена на рис. 5.

Естественный параллелизм СОК позволяет эффективно задействовать ресурсы как CPU, так и GPU. При использовании GPU сразу n потоков обрабатывают цифры мантиссы, при этом каждый i-й GPU-поток выполняет операции над остатком x_i по i-му модулю m_i. Вычисления с ИПХ выполняются параллельно с вычислениями в СОК с использованием формул интервальной арифметики. Анализ ИПХ проводится только тогда, когда нужно оценить мантиссу, к примеру, для контроля переполнения или определения знака. В этом случае вводятся точки синхронизации потоков. При необходимости ИПХ "обновляется", т. е. пересчитывается на основании цифр мантиссы. Для этого разработан специальный алгоритм [24], который хорошо распараллеливается.

В случае вычислений на CPU циклы обработки мантисс и вычисления ИПХ эффективно векторизуются с использованием SIMD-расширений. Кроме этого, при большом числе модулей и низком уровне параллелизма решаемой задачи может оказаться целесообразной многопоточная CPU-реализация, при которой цифры мантисс обрабатываются группами на различных ядрах универсального процессора.

Modular-Positional Floating-Point. Данный формат чисел (сокращенно MF-формат) расширяет описанный выше MI-формат посредством добавления несмещенного порядка (экспоненты) **e**, который представлен целым машинным числом со знаком. Для записи чисел используется нотация $x \to \{s, X, \mathbf{e}, I(X/M)\}$, соответствующая значению

$$x = \left(-1^s\right) \times \left| \sum_{i-1}^{n} x_i \mid M_i^{-1} \mid_{m_i} M_i \right|_M \times 2^{\mathbf{e}}.$$

В MF-формате определены конечные числа (включая знаковые нули), бесконечности со знаком и

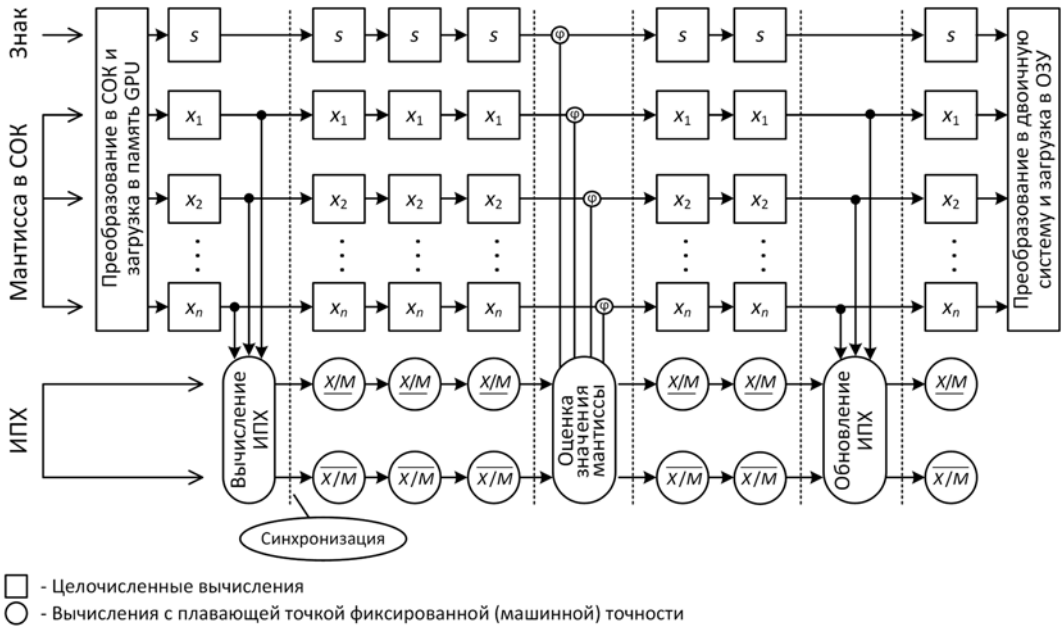


Рис. 5. Схема многоразрядных вычислений в MI-формате на GPU; функция φ вычисляет цифру мантиссы на основании результата немодульной операции

```
/*Internal structure*/
typedef struct {
     mf_sig_t s;                    //Sign
     mf_exp_t exp;                  //Exponent
     mf_res_t residue[MODULI_SIZE]; //Significand in RNS
     er_t     ipc_low;              //Lower IPC bound
     er_t     ipc_upp;              //Upper IPC bound
} __mp_float_struct;

/*User data types*/
typedef __mp_float_struct *mf_t;           //For single number
typedef __mp_float_struct *mf_arr_t;       //For arrays
typedef __mp_float_struct  mf_static_t;    //For device side code
```

Рис. 6. Типы данных с плавающей точкой произвольной длины

специальные нечисловые данные (*Not-a-Number*). Как и в стандарте IEEE-754, бесконечности интерпретируются в аффинном смысле. На рис. 6 представлены типы данных, реализующие MF-формат. Основные алгоритмы обработки чисел в MF-формате изложены в работах [21, 25].

Структура и функциональные возможности

Пакет MPRES состоит из двух идентичных по функциональным возможностям модулей — CPU и GPU. Оба модуля используют одинаковые типы данных, что обеспечивает их совместимость. Каждый модуль разбит на три блока: блок арифметики расширенного диапазона, блок целочисленной многоразрядной арифметики и блок арифметики с плавающей точкой многократной точности. Внутри каждого блока выполняется разделение по типам выполняемых операций (рис. 7).

Все процедуры в MPRES представлены в CPU- и в GPU-реализациях, которые могут использоваться одновременно. Поддерживается концепция потоковой безопасности. Базовая функциональность включает подпрограммы выполнения арифметических операций над рассмотренными выше классами чисел. Для основных арифметиче-

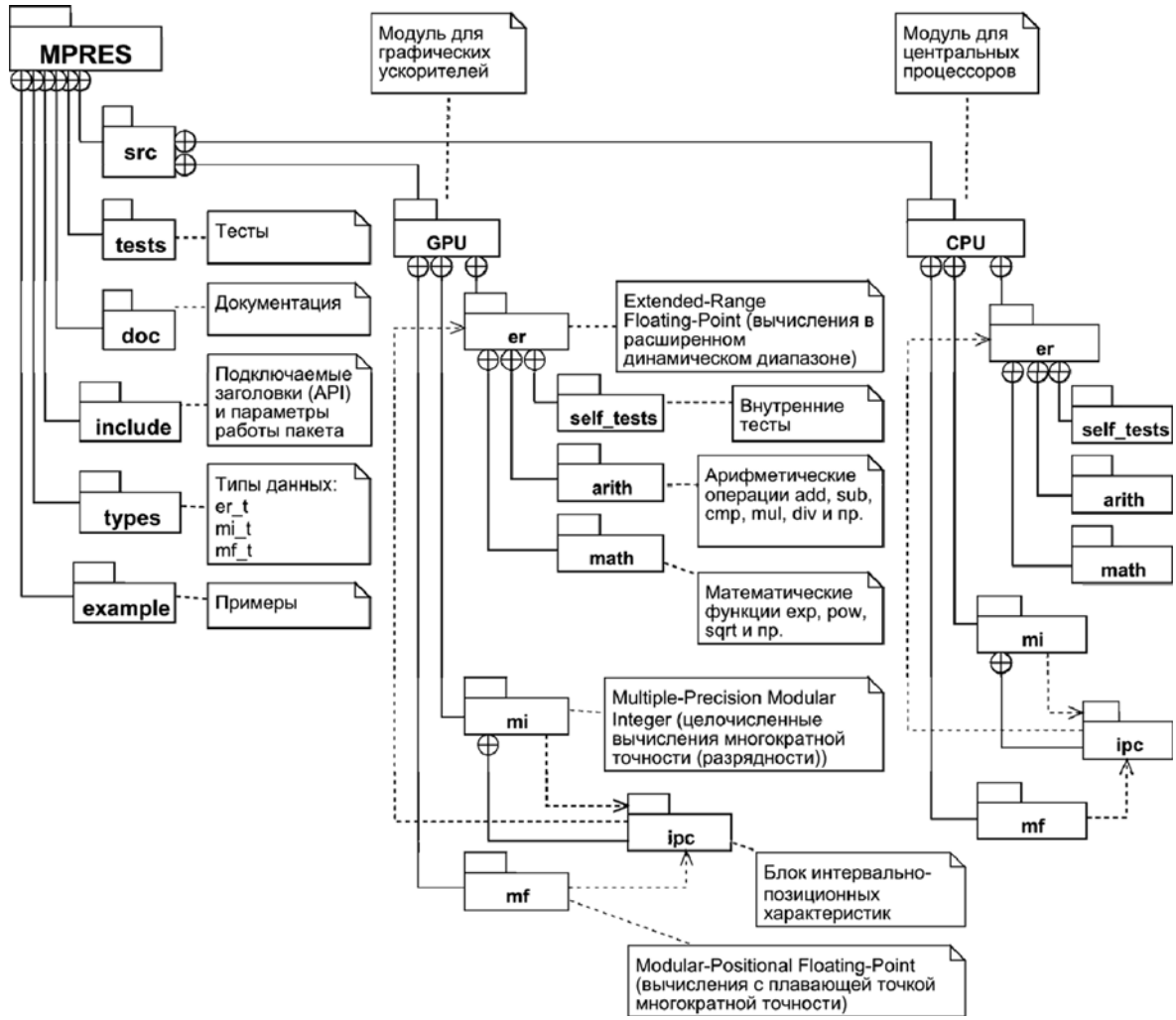


Рис. 7. Диаграмма пакетов MPRES с детализацией блока арифметики расширенного диапазона (`er`)

ских операций в формате с расширенной экспонентой реализована поддержка четырех режимов округления IEEE-754: до ближайшего (*round to nearest*), вверх (*round toward* $+\infty$), вниз (*round toward* $-\infty$) и к нулю (*round toward zero*), а также ряд других математических функций: exp, fact, pow, sqrt, sin, cos, ceil, floor и пр. Для всех трех типов данных предусмотрены функции преобразования из стандартных (машинных) форматов и обратно.

Для расширения возможностей MPRES дополнительно будут реализованы: параллельные операции над массивами целых чисел произвольной длины; высокоточное вычисление алгебраических и трансцендентных (логарифмических и показательных, тригонометрических, гиперболических и соответствующих им обратных) функций (для чисел с плавающей точкой); оптимизированные примитивы для хранения и эффективной обработки плотных и разреженных матриц, параллельные методы выполнения базовых операций линейной алгебры повышенной точности: GEMV, GEMM и другие BLAS-операции; высокоточные параллельные методы решения плохо обусловленных линейных систем больших порядков. Запланирована поддержка динамического изменения точности вычислений, реализация которой требует разработки эффективных методов расширения оснований СОК.

Заключение

Представлены результаты разработки потокобезопасного программного пакета MPRES, ориентированного на системы с гибридными узлами из центральных процессоров и графических ускорителей, поддерживающих CUDA. Этот пакет реализует два формата данных многократной точности (целочисленный и с плавающей точкой), а также вычисления фиксированной точности в расширенном динамическом диапазоне. В основе высокоточных модулей пакета лежит использование систем остаточных классов для представления многоразрядных мантисс чисел, что допускает эффективное распараллеливание высокоточных операций (путем одновременной обработки сразу всех цифр мантиссы) и позволяет преодолеть основное ограничение позиционных систем счисления, связанное с образованием цепочек переносов между разрядами.

На настоящее время завершена работа над модулем арифметики расширенного диапазона, разрабатываются алгоритмы и подпрограммы модулей многократной точности. Кроме многоядерных CPU и CUDA-совместимых GPU, в перспективе запланирована поддержка других параллельных вычислительных архитектур, таких как Intel MIC.

Работа выполнена при финансовой поддержке РФФИ в рамках научного проекта № 16-37-60003 мол_а_дк.

Список литературы

1. **Viswanath D., Sahutŏglu S.** Complex Singularities and the Lorenz Attractor // SIAM Review. 2010. Vol. 52. P. 294—314.

2. **Leweke S., von Lieres E.** Fast Arbitrary Order Moments and Arbitrary Precision Solution of the General Rate Model of Column Liquid Chromatography With Linear Isotherm // Computers & Chemical Engineering. 2016. Vol. 84. P. 350—362.

3. **Laskar J., Gastineau M.** Existence of collisional trajectories of Mercury, Mars and Venus with the Earth // Nature. 2009. Vol. 459, N 7248. P. 817—819.

4. **He Y., Ding C.** Using Accurate Arithmetics to Improve Numerical Reproducibility and Stability in Parallel Applications // The Journal of Supercomputing. 2001. Vol. 18. P. 259—277.

5. **Granlund T.** The GNU Multiple Precision Arithmetic Library, Edition 6.1.0. 2015. 148 p. URL: https://gmplib.org/gmp-man-6.1.0.pdf (date of access 22.05.2016).

6. **Fousse L., Hanrot G., Lefèvre V., Pélissier P., Zimmermann P.** MPFR: A Multiple-Precision Binary Floating-Point Library with Correct Rounding // ACM Transactions on Mathematical Software. 2007. Vol. 33, N 2. Art. 13.

7. **Mehlhorn K., Näher St.** LEDA: A Platform for Combinatorial and Geometric Computing. Cambridge, UK: Cambridge University Press, 1999.

8. **Bailey D. H., Hida Y., Li X. S., Thompson B.** ARPREC: An Arbitrary Precision Computation Package. 2002. 8 p. URL: http://www.davidhbailey.com/dhbpapers/arprec.pdf (date of access 15.03.2016).

9. **Owens J. D., Luebke D., Govindaraju N. K.** et al. A Survey of General-Purpose Computation on Graphics Hardware // Computer Graphics Forum. 2007. Vol. 26, N 1. P. 80—113.

10. **Zhao K., Chu X.** GPUMP: A Multiple-Precision Integer Library for GPUs // Proceedings of 10th International Conference on Computer and Information Technology. Bradford, UK: IEEE. 2010. P. 1164—1168.

11. **Nakayama T., Takahashi D.** Implementation of Multiple-Precision Floating-Point Arithmetic Library for GPU Computing // Proceedings of 3rd Int. Conf. on Parallel and Distributed Computing and Systems. Dallas, USA. 2011. P. 343—349.

12. **Lu M., He B., Luo Q.** Supporting Extended Precision on Graphics Processors // Proceedings of 6th International Workshop on Data Management on New Hardware. NY, USA: ACM. 2010. P. 19—26.

13. **Szabo N., Tanaka R.** Residue Arithmetic and its Application to Computer Technology. NY, USA: McGraw-Hill, 1967.

14. **Parhami B.** Computer Arithmetic: Algorithms and Hardware Designs. NY, USA: Oxford University Press, 2000.

15. **Акушский И. Я., Юдицкий Д. И.** Машинная арифметика в остаточных классах. М.: Сов. радио, 1968. 440 с.

16. **Albicocco P., Cardarilli G., Nannarelli A., Re M.** Twenty years of research on RNS for DSP: Lessons learned and future perspectives // Proceedings of 14th Int. Symp. Integrated Circuits (ISIC). Singapore. 2014. P. 436—439.

17. **Esmaeildoust M., Schinianakis D., Javashi H., Stouraitis T., Navi K.** Efficient RNS implementation of elliptic curve point multiplication over GF(p) // IEEE Transactions on VLSI Systems. 2013. Vol. 21, N 8. P. 1545—1549.

18. **Goh V. T., Siddiqi M. U.** Multiple error detection and correction based on redundant residue number systems // IEEE Transactions on Communications. 2008. Vol. 56, N 3. P. 325—330.

19. **Wang W., Swamy M., Ahmad M.** RNS application for digital image processing // Proceedings of 4th IEEE Int. Workshop Syst.-on-Chip for Real Time Applications. Banff, Alberta, Canada. 2004. P. 77—80.

20. **Исупов К. С., Князьков В. С.** Немодульные вычисления в системах остаточных классов с интервально-позиционными характеристиками. Киров, 2015. 92 с. Деп. в ВИНИТИ РАН 26.03.2015, № 61-В2015.

21. **Isupov K., Knyazkov V.** A Modular-Positional Computation Technique for Multiple-Precision Floating-Point Arithmetic // Lecture Notes in Computer Science. 2015. Vol. 9251. P. 47—61.

22. **Hauser J. R.** Handling Floating-Point Exceptions in Numeric Programs // ACM Transactions on Programming Languages and Systems. 1996. Vol. 18, N 2. P. 139—174.

23. **Виноградов И. М.** Основы теории чисел. М.: Государственное изд-во технико-теоретической литературы, 1952. 180 с.

24. **Исупов К. С.** Об одном алгоритме сравнения чисел в системе остаточных классов // Вестник АГТУ. Серия "Управление, вычислительная техника и информатика". 2014. N 3. С. 40—49.

25. **Исупов К. С., Князьков В. С.** Арифметика многократной точности на основе систем остаточных классов // Программные системы: теория и приложения. 2016. Т. 7, № 1 (28). С. 61—97.

Development of High-Precision Arithmetic Package for Supercomputers with Graphics Processing Units

K. S. Isupov, ks_isupov@vyatsu.ru, **V. S. Knyazkov,** knyazkov@vyatsu.ru,
A. S. Kuvaev, kyvaevy@gmail.com, **M. V. Popov,** mihailpopov1993@mail.ru,
Vyatka State University, Kirov, 610000, Russian Federation

Corresponding author:

Isupov Konstantin S., Leading Researcher, Vyatka State University, 610000, Kirov, Russian Federation, e-mail: ks_isupov@vyatsu.ru

Received on May 30, 2016
Accepted on July 06, 2016

A multiple-precision software library for parallel computing systems with hybrid CPU/GPU architecture is considered in this paper. Three numeric data types are supported: fixed-precision extended-range floats, multiple-precision integers, and multiple-precision floats. The feature of the library is the use of residue number system (RNS) to represent multiple-precision significands. This makes it possible to effective parallel processing of significand digits due to the lack of carry propagation. The interval estimation of relative (fractional) values in RNS is used to perform complex operations, such as magnitude comparison and overflow detection.

Keywords: *multiple-precision computations, extended range, residue number system, CUDA, software*

Acknowledgements: *This work was supported by the Russian Foundation for Basic Research, project no. 16-37-60003 mol_a_dk*

For citation:

Isupov K. S., Knyazkov V. S., Kuvaev A. S., Popov M. V. Development of High-Precision Arithmetic Package for Supercomputers with Graphics Processing Units, *Programmnaya Ingeneria*, 2016, vol. 7, no. 9, pp. 387—394.

DOI: 10.17587/prin.7.387-394

References

1. **Viswanath D., Sahutoglu S.** Complex Singularities and the Lorenz Attractor, *SIAM Review*, 2010, vol. 52, pp. 294—314.
2. **Leweke S., von Lieres E.** Fast Arbitrary Order Moments and Arbitrary Precision Solution of the General Rate Model of Column Liquid Chromatography With Linear Isotherm, *Computers & Chemical Engineering,* 2016, vol. 84, pp. 350—362.
3. **Laskar J., Gastineau M.** Existence of collisional trajectories of Mercury, Mars and Venus with the Earth, *Nature,* 2009, vol. 459, no. 7248, pp. 817—819.
4. **He Y., Ding C.** Using Accurate Arithmetics to Improve Numerical Reproducibility and Stability in Parallel Applications, *The Journal of Supercomputing,* 2001, vol. 18, pp. 259—277.
5. **Granlund T.** *The GNU Multiple Precision Arithmetic Library*, Edition 6.1.0. 2015. 148 p., available: https://gmplib.org/gmp-man-6.1.0.pdf (date of access 22.05.2016).
6. **Fousse L., Hanrot G., Lefèvre V., Pélissier P., Zimmermann P.** MPFR: A Multiple-Precision Binary Floating-Point Library with Correct Rounding, *ACM Transactions on Mathematical Software*, 2007, vol. 33, no. 2, Art. 13.
7. **Mehlhorn K., Näher St.** *LEDA: A Platform for Combinatorial and Geometric Computing,* Cambridge, UK: Cambridge University Press, 1999.
8. **Bailey D. H., Hida Y., Li X. S., Thompson B.** ARPREC: An Arbitrary Precision Computation Package. 2002. 8 p., available: http://www.davidhbailey.com/dhbpapers/arprec.pdf (date of access 15.03.2016).
9. **Owens J. D., Luebke D., Govindaraju N. K., Harris M., Kruger J., Lefohn A. E., Purcell T. J.** A Survey of General-Purpose Computation on Graphics Hardware, *Computer Graphics Forum*, 2007, vol. 26, no. 1, pp. 80—113.
10. **Zhao K., Chu X.** GPUMP: A Multiple-Precision Integer Library for GPUs, *Proceedings of 10th International Conference on Computer and Information Technology*, Bradford, UK: IEEE, 2010, pp. 1164—1168.
11. **Nakayama T., Takahashi D.** Implementation of Multiple-Precision Floating-Point Arithmetic Library for GPU Computing, *Proceedings of 3rd Int. Conf. on Parallel and Distributed Computing and Systems*, Dallas, USA, 2011, pp. 343—349.
12. **Lu M., He B., Luo Q.** Supporting Extended Precision on Graphics Processors, *Proceedings of 6th International Workshop on Data Management on New Hardware*, NY, USA: ACM, 2010, pp. 19—26.
13. **Szabo N., Tanaka R.** *Residue Arithmetic and its Application to Computer Technology,* NY, USA, McGraw-Hill, 1967.
14. **Parhami B.** *Computer Arithmetic: Algorithms and Hardware Designs*, NY, USA, Oxford University Press, 2000.
15. **Akushskij I. Ja., Judickij D. I.** *Mashinnaja arifmetika v ostatochnyh klassah* (Machine arithmetic in residual classes), Moscow, Sov. Radio, 1968, 440 p. (in Russian).
16. **Albicocco P., Cardarilli G., Nannarelli A., Re M.** Twenty years of research on RNS for DSP: Lessons learned and future perspectives, *Proceedings of 14th Int. Symp. Integrated Circuits (ISIC),* Singapore, 2014, pp. 436—439.
17. **Esmaeildoust M., Schinianakis D., Javashi H., Stouraitis T., Navi K.** Efficient RNS implementation of elliptic curve point multiplication over GF(p), *IEEE Transactions on VLSI Systems*, 2013, vol. 21, no 8, pp. 1545—1549.
18. **Goh V. T., Siddiqi M. U.** Multiple error detection and correction based on redundant residue number systems, *IEEE Transactions on Communications*, 2008, vol. 56, no. 3, pp. 325—330.
19. **Wang W., Swamy M., Ahmad M.** RNS application for digital image processing, *Proceedings of 4th IEEE Int. Workshop Syst.-on-Chip for Real Time Applications*, Banff, Alberta, Canada, 2004, pp. 77—80.
20. **Isupov K. S., Knjaz'kov V. S.** *Nemodul'nye vychislenija v sistemah ostatochnyh klassov s interval'no-pozicionnymi harakteristikami* (Non-modular computations in residue number systems with interval-positional characteristics), Kirov, 2015, 92 p., Dep. v VINITI RAN 26.03.2015, no. 61-V2015 (in Russian).
21. **Isupov K., Knyazkov V.** A Modular-Positional Computation Technique for Multiple-Precision Floating-Point Arithmetic, *Lecture Notes in Computer Science*, 2015, vol. 9251, pp. 47—61.
22. **Hauser J. R.** Handling Floating-Point Exceptions in Numeric Programs, *ACM Transactions on Programming Languages and Systems*, 1996, vol. 18, no. 2, pp. 139—174.
23. **Vinogradov I. M.** *Osnovy teorii chisel* (Fundamentals of Number Theory), Moscow, Gosudarstvennoe izd-vo tehniko-teoreticheskoj literatury, 1952, 180 p. (in Russian).
24. **Isupov K. S.** Ob odnom algoritme sravnenija chisel v sisteme ostatochnyh klassov(On an algorithm for number comparison in the residue number system), *Vestnik AGTU. Serija "Upravlenie, vychislitel'naja tehnika i informatika"*, 2014, no. 3, pp. 40—49 (in Russian).
25. **Isupov K. S., Knjaz'kov V. S.** Arifmetika mnogokratnoj tochnosti na osnove sistem ostatochnyh klassov (Parallel multiple-precision arithmetic based on residue number), *Programmnye sistemy: teorija i prilozhenija*, 2016, vol. 7, no. 1(28), pp. 61—97 (in Russian).

УДК 004.05 DOI: 10.17587/prin.7.395-399

В. Е. Гвоздев, д-р техн. наук, проф., зав. каф., e-mail:wega55@mail.ru,
Д. В. Блинова, канд. техн. наук, доц.,e-mail: blinova.darya@gmail.com,
Уфимский государственный авиационный технический университет

Анализ функциональных возможностей аппаратно-программных комплексов на ранних стадиях проектирования с учетом мнений правообладателей*

Предложена векторная модель, характеризующая полезные и вредные функции аппаратно-программных комплексов при различных режимах использования последних. Представлена многомерная модель, позволяющая анализировать свойства аппаратно-программных комплексов с учетом различия в точках зрения правообладателей на функциональные возможности объекта. Использование предложенных моделей позволяет повысить степень формализации описания функциональных требований на ранних стадиях проектирования, что делает возможным разработку формальных процедур оценки соответствия свойств базового объекта желаниям и ожиданиям неоднородных правообладателей. Приведены определения таких понятий, как "базовый образец", "эталонный образец", "дефект". Предложена процедура оценки значимости проявления дефектов по совокупности оценок группы правообладателей, основанная на использовании функций принадлежности.

Ключевые слова: *дефект, дефектология, правообладатель, функция принадлежности, аппаратно-программный комплекс, вредная функция, полезная функция*

Введение

Постоянно возрастающая роль систем обработки информации и управления в жизни современного общества привела к появлению нового направления научных исследований и создания прикладных инструментальных средств — дефектологии аппаратно-программных комплексов (АПК) [1—3]. К числу опорных понятий в рамках этого направления относятся "качество" и "симптомы дефектов".

Понятие "дефект" неразрывно связано с понятием "базовый образец" [4], "эталонный образец". Такая связь обусловлена тем обстоятельством, что о проявлении дефекта можно говорить лишь в том случае, если определены базовые, либо эталонные свойства изделия. Базовые свойства соответствуют консолидированному мнению правообладателей, имеющих разные представления (мнения) о желаемых свойствах проектируемого изделия, представленны в официально утвержденных документах [4]. Эталонные свойства соответствуют субъективному представлению отдельного правообладателя о желаемых свойствах изделия и не представлены в каких-либо официальных документах.

Основу оценки эффективности изделий, в которых средства передачи информации и обработки данных играют определяющую роль, составляет соответствие функциональных возможностей изделий субъективным требованиям и ожиданиям правообладателей. Это обусловлено тем, что если функциональные возможности объекта не соответствуют потребностям, желаниям и ожиданиям правообладателей, то такой объект не представляет интереса. Основу оценивания значимости дефектов составляет квалиметрия.

Разработка подходов к анализу симптомов дефектов, характеризуемых степенью отклонения показателей качества объекта от базовых, определенных в техническом задании на его разработку, позволит создать информационную основу для принятия решения об обоснованности действий по локализации и устранению дефектов и об объемах ресурсов, которые следует выделить для снижения количества дефектов до уровня, приемлемого с точки зрения пользователя изделия [5, 6].

Вредные и полезные функции АПК

В качестве концептуальной основы построения эталонных/базовых моделей изделий предлагается использовать функциональный поход. Это обусловлено тем, что рассмотрение АПК на уровне

* Понятие "правообладатель" понимается в смысле, приведенном в ГОСТ Р ИСО/МЭК 15288-2005 — "Информационная технология. Системная инженерия. Процессы жизненного цикла систем".

функциональных свойств позволяет сравнивать между собой изделия разной структуры, созданные на разной элементной базе. Функциональный подход создает единую платформу для сравнения разных частей изделий.

Одним из базовых условий построения эталонных/базовых моделей, по мнению авторов, должен быть учет того, что при функционировании сложных систем разной природы и различного назначения имеет место одновременная реализация как полезных (*useful*), так и вредных (*harmful*) функций [7, 8]. В дальнейшем под полезной функцией АПК понимается такая функция, которая увеличивает ценность продукта (услуги), поставляемого пользователю, либо уменьшает затраты, связанные с получением ценного результата.

Под вредной функцией понимается такая функция, которая уменьшает ценность продукта (услуги), поставляемого пользователю, либо увеличивает затраты, связанные с получением ценного результата.

Заранее заданное предельное соотношение между полезными и вредными функциями определяет границы функциональной пригодности изделий. Число полезных и вредных функций, реализуемых системой, зависит, во-первых, от требований целевых групп правообладателей, во-вторых, от режима использования изделия (состояния окружающей среды).

В разных режимах использования ε_k $\left(k = \overline{1;\ K}\right)$ объект может реализовывать с точки зрения пользователей различное число полезных и вредных функций. При этом отнесение одной и той же функции к классу полезных либо вредных зависит от режима использования и ценностных установок пользователей (внешних правообладателей). Предполагая число режимов использования заранее определенным и равным K, можно определить полное множество функций F_Σ, реализуемое объектом во всех режимах:

$$F_\Sigma = \bigcup_{k=1}^{K} \left\{F^{(\varepsilon_k)}\right\},$$

где $\left\{F^{(\varepsilon_k)}\right\}$ — множество функций, реализуемых изделием в режиме ε_k, т. е. F_Σ определяет границы качества АПК в рамках функционального подхода [9].

Пронумеровав элементы множества F_Σ, каждому режиму ε_k с учетом ценностных установок m-й группы внешних правообладателей $\left(m = \overline{1;\ M}\right)$ можно поставить в соответствие вектор вида

$$\mathbf{L} = (1,\ 0,\ -1,\ ...,\ 0,...,1). \qquad (1)$$

Число компонент вектора **L** совпадает с мощностью множества F_Σ. Компоненты вектора имеют следующий смысл: "1" - функция реализуется в объеме, оговоренном в техническом задании (т. е. в объеме, соответствующем базовому изделию), и является для m-й группы правообладателей полезной; "0" — функция не реализуется; "−1" — функция реализуется в объеме, определенном в техническом задании, и для m-й группы правообладателей является вредной.

Множеству режимов использования изделия $\{\varepsilon_k\}_1^K$, множеству групп правообладателей $\{m\}_1^K$, множеству функций F_Σ с учетом выражения (1) можно поставить в соответствие многомерную модель, представленную на рис. 1.

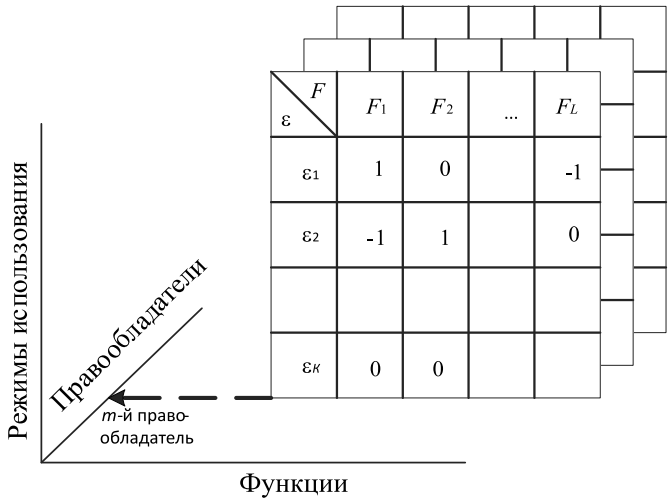


Рис. 1. Многомерное представление эталонных моделей

Преобразуем выражение (1) к виду

$$\left(\alpha_1^{(\varepsilon_k)(m)},\ \alpha_2^{(\varepsilon_k)(m)},\ ...,\ \alpha_L^{(\varepsilon_k)(m)}\right),$$

где $\alpha_l^{(\varepsilon_k)(m)}$ — показатель (вес), характеризующий полноту реализации l-й функции относительно функции, реализуемой базовым образцом в режиме ε_k с точки зрения m-го правообладателя, $\alpha_l^{(\varepsilon_k)(m)} \in \left[-1;\ \infty\right)$.

Заметим, что показатель $\alpha_l^{(\varepsilon_k)(m)}$ может отличаться от эталонного значения как в большую, так и в меньшую сторону.

Отклонение в меньшую сторону свидетельствует о наличии дефектов, возникающих при переводе требований пользователей в программные продукты и конфигурацию АПК. Нижняя граница $\alpha_l^{(\varepsilon_k)(m)}$ подчеркивает то обстоятельство, что наличие непреднамеренного дефекта может сделать полезную функцию вредной. Отклонения в большую сторону соответствует наличию функциональных возможностей, превышающих потребности пользователей, иными словами, свидетельствуют о наличии организационных дефектов программного проекта: бюджет проекта тратится на то, чего заказчик не требует.

Анализ качества базового образца с использованием функций принадлежности

В работе [10] отмечается, что примерно 50 % функциональных возможностей систем обработки данных оказываются невостребованными пользователями. Изучение мнения правообладателей о ценностях функций, соответствующих базовому образцу, позволит на ранних стадиях проектирования обоснованно скорректировать внешний облик объекта. Это означает устранение фундаментальных

дефектов проектирования, следствием которых является наличие у объекта свойств, малозначимых с точки зрения большинства правообладателей.

В работах [5, 11, 12] и других литературных источниках, посвященных проблематике управления рисками, отмечается невозможность выявления всех дефектов и нецелесообразность устранения всех выявленных дефектов. Не выявленные, либо выявленные, но не устраненные дефекты являются причиной отклонения свойств реального объекта от свойств базового, либо эталонного образца. Это выражается в различных искажениях выходных результатов относительно ожидаемых правообладателями, т. е. в форме "симптомов дефектов".

В силу различия субъективного представления неоднородных правообладателей о свойствах эталонного образца, отношение разных правообладателей к одним и тем же симптомам дефектов будет различным. Рассмотрим формальную процедуру анализа значимости дефектов на основе оценки правообладателями результатов подконтрольной эксплуатации объекта. Основу процедуры составляет технология с использованием функций принадлежности, описанная в работах [13, 14].

Сформируем набор значений лингвистической шкалы: "несущественные искажения результатов", "приемлемые искажения результатов", "недопустимые искажения результатов". Содержание этих шкал определим по аналогии с работой [5]. Следует отметить, что в упомянутом источнике используется иная терминология: "небольшие ошибки", "умеренные ошибки", "критические ошибки". По мнению авторов, употребляемые в работе [5] понятия требуют дополнительного обсуждения. Однако это исследование выходит за рамки настоящей статьи. Поэтому пока установим следующие аналогии: "несущественные искажения результатов" — "небольшие ошибки", "приемлемые искажения результатов" — "умеренные ошибки", "недопустимые искажения результатов" — "критические ошибки".

Несущественные искажения результатов — такие, на которые средний пользователь не обратит внимания и последствия которых так и не обнаружатся (например, орфографические ошибки на экране). По десятибалльной шкале возможного ущерба от неисправности, погрешности или неумышленного искажения объекта или процесса (в силу дефектов в интерфейсах и модели объекта/процесса) таким искажениям присваивается значение в диапазоне от 1 до 3 баллов.

Приемлемые искажения результатов — такие, которые влияют на конечного пользователя, но имеются слабые последствия или обходные пути, позволяющие сохранить достаточную функциональность изделия (например, ошибочный текст на экране; сбои, если их трудно воспроизвести и они не оказывают влияния на существенное число пользователей). По десятибалльной шкале возможного ущерба этому классу присваивается значение в диапазоне от 4 до 7 баллов.

Недопустимые искажения результатов приводят к прекращению использования изделия. По десятибалльной шкале возможного ущерба этому классу присваивается значение в диапазоне от 8 до 10 баллов.

Каждому q-му значению предлагаемой шкалы присваивается своя нечеткая функция принадлежности μ_q, определяемая на балльной оси [15]. Положение максимума q-й функции принадлежности является его опорным значением r_q. В случае если максимальному значению соответствует подынтервал балльной шкалы, опорным значением является середина подынтервала. Для описанной выше лингвистической шкалы предлагаются функции принадлежности (рис. 2).

В терминах лингвистической шкалы m-й правообладатель выражает свое мнение о результатах реализации каждой из функций, входящих в подмножества вредных и полезных функций, которые соотносятся с k-м режимом использования (см. рис. 1). Помимо этого, правообладатель указывает уверенность в выбранном значении степени искажения результатов l-й функции по сравнению с эталонными, выраженную числом $\mu_l \in [0; 1]$. В расчет принимаются лишь полезные/вредные функции, т. е. такие, которым в модели базового образца (см. рис. 1) в k-й строке ставятся в соответствие значения "+1", "−1".

Для каждого из режимов ε_k с учетом оценки качества реализации l-й функции с точки зрения m-го правообладателя рассчитывается скалярная величина, дающая совокупную оценку значимости дефектов:

$$Z^{(k),(m)} = \frac{\sum_{l=1}^{L_k^{*(k),(m)}} r_l^{(m)} \mu_l^{(k),(m)}}{\sum_{l=1}^{L^*} \mu_l^{(k),(m)}}, \qquad (2)$$

где $L_k^{*(k),(m)}$ — число ненулевых ячеек в k-й строке m-й матрицы (см. рис. 2); $r_l^{(m)}$ — опорное значение, соответствующее лингвистической оценке l-й функции, данной m-й правообладателем по результатам подконтрольной эксплуатации.

Если все значения $r_l^{(m)}$ в выражении (2) одинаковы, то значение $\mu_l^{(k),(m)}$ не играет роли. В этом случае степень уверенности полагается равной [13, 14]

$$\mu_l^{*(k),(m)} = \min\left\{\mu_1^{(k),(m)},\ \mu_2^{(k),(m)},\ \ldots,\ \mu_{L^{*(k),(m)}}^{(k),(m)}\right\}.$$

Величина $Z^{(k),(m)}$ является аналогом "среднего риска" и может выступать в качестве комплексной характеристики качества изделий некритического назначения (например, бытовой техники).

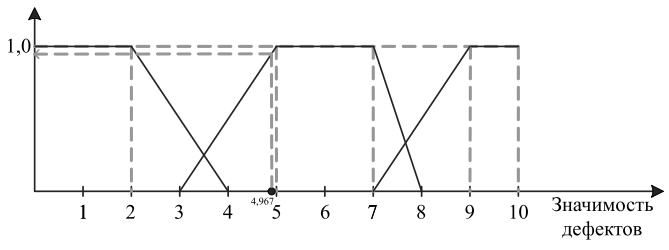


Рис. 2. Функция принадлежности лингвистической шкалы

Пример. Предположим, в k-м режиме изделием реализуется одна полезная функция и одна вредная функция, т. е. $L^{*(k),(m)} = 2$. Правообладатель оценивает результаты реализации функции следующим образом:

F_1: {"несущественное искажение результатов"; $\mu_1 = 0{,}8$},

F_2: {"недопустимое искажение результатов"; $\mu_2 = 0{,}7$}.

Опорное значение для лингвистической переменной "несущественное искажение результатов" составляет $r_1 = 1$ балл. Опорное значение для лингвистической переменной "недопустимое искажение результатов" составляет $r_2 = 9{,}5$ баллов (см. рис. 2). Согласно формуле (2)

$$Z^{(k),(m)} = \frac{1 \cdot 0{,}8 + 9{,}5 \cdot 0{,}7}{0{,}8 + 0{,}7} = 4{,}967.$$

Ближайшее к числу 4,967 опорное значение равно шести, чему соответствует лингвистическая переменная "приемлемое искажение результатов". Для $Z^{(k),(m)} = 4{,}967$ значение функции принадлежности $\mu_l^{(k),(m)}$ практически равно единице. Таким образом, по совокупности результатов эксплуатации m-й правообладатель оценивает функционирование системы как "приемлемое искажение результатов" фактически со 100 %-ной уверенностью.

Полученные результаты позволяют заключить, что в целом изделие представляет для пользователя интерес, однако часть функций либо требует серьезной доработки, либо эти функции следует исключить.

Следует подчеркнуть, что в рассмотренном примере значимость обеих функций для правообладателя считалась одинаковой.

Оценка (2) не может использоваться для систем критического назначения. В этом случае в качестве $Z^{(k),(m)}$ следует использовать лингвистическую переменную, соответствующую опорному значению

$$r^{(k),(m)} = \max\{r_l\}_1^{L^*(k),(m)},$$

т. е. давать заниженную, осторожную оценку. Это соответствует тому, что с точки зрения m-го правообладателя в изделии присутствуют значительные дефекты, исключающие возможность его использования в k-м режиме.

Заключение

В рамках функционального подхода предложена формальная модель описания базовых свойств аппаратно-программных комплексов с учетом наличия вредных и полезных функций. Предлагаемая модель в отличие от известных подходов к формированию спецификации требований пользователей позволяет повысить степень формализации описания функциональных требований на ранних стадиях реализации проектов, что делает возможным разработку формальных процедур оценки соответствия свойств базового объекта желаниям и ожиданиям неоднородных правообладателей. Предложено использование функции принадлежности лингвистической шкалы для оценки качества базового образца с точки зрения правообладателя.

Работа поддержана грантом РФФИ 16-08-00442 "Управление функциональной безопасностью аппаратно-программных комплексов в составе сложных технических систем".

Список литературы

1. **Марков А. С., Фадин А. А.** Систематика уязвимостей и дефектов безопасности программных ресурсов // Защита информации INSIDE. 2013. № 3. С. 2—7.

2. **Бородакий Ю. В., Юсупов Р. М., Пальчун Б. П.** Проблема имитационного моделирования дефектоскопических свойств компьютерной инфосферы // Труды третьей всероссийской научно-практической конференции "Имитационное моделирование. Теория и практика". Санкт-Петербург, 2007. С. 87—92.

3. **Chai K. H., Zhang J., Tan K. C.** A TRIZ-Based Method for New Service Design. National University of Singapore // Journal of Service Research. 2005. Vol. 8, N 1. P. 48—66.

4. **ГОСТ Р ИСО/МЭК 15288—2005** Информационная технология. Системная инженерия. Процессы жизненного цикла систем.

5. **Липаев В. В.** Анализ и сокращение рисков проектов сложных программных средств. М.: СИНТЕГ, 2005. 224 с.

6. **Гвоздев В. Е., Блинова Д. В.** Онтологический анализ дефектов при проектировании компонентов аппаратно-программных комплексов // Онтология проектирования. 2015. Том 5, № 4 (18). С. 399—410.

7. **Лапыгин Ю. Н.** Стратегический менеджмент: Учеб. пособие. М.: Высшее образование, 2007. 174 с.

8. **Гвоздев В. Е., Блинова Д. В.** Элементы теории управления непреднамеренными дефектами в информационно-коммуникационных системах // Системная инженерия. 2015. № 2. С. 104—113.

9. **Чекмарев А. И.** Квалиметрия и управление качеством. Часть 1. Квалиметрия: Учеб. пособие. Самара: СГАУ, 2010. 172 с.

10. **CHAOS** Manifesto 2011: The Laws of CHAOS and the CHAOS 100 Best PM Practices. The Standish Group International, Incorporated.

11. **Майерс Г.** Надежность программного обеспечения. М.: Мир, 1980. 360 с.

12. **Черкесов Г. Н.** Надежность аппаратно-программных комплексов: Учеб. пособие. СПб.: Питер, 2005. 479 с.

13. **Pelaez C. E., Bowles J. B.** Using fuzzy cognitive maps as a system model for failure modes and effects analysis // Information Sciences. 1996. N 88 (1). P. 177—199.

14. **Кузнецов О. П., Кулинич А. А., Марковский А. В.** Анализ влияний при управлении слабоструктурированными ситуациями на основе когнитивных карт // Человеческий фактор в управлении. М.: КомКнига, 2006. С. 313—344.

Analysis of Functional Possibilities of Hardware-Software Complexes in the Early Stages of Designing with Considering the Opinions of Stakeholders

V. E. Gvozdev, wega55@mail.ru, **D. V. Blinova,** blinova.darya@gmail.com, Ufa State Aviation Technical University, Ufa, 450044, Russian Federation

Corresponding author:

Blinova Darya V., Assistant Professor, Ufa State Aviation Technical University, Ufa, 450044, Russian Federation, e-mail: blinova.darya@gmail.com

Received on May 12, 2016
Accepted on June 30, 2016

The paper discusses the functionality of the software-hardware complexes from the point of view of different stakeholders.

A vector model which characterizes the useful and harmful functions of the hardware-software complexes for different modes of use is proposed. The vector components characterize matching functionality and basic properties of the hardware-software systems. Basic properties are understood as the properties corresponding to the object's base line.

Proposed is the multivariate model, which allows one to analyze the properties of hardware-software complexes taking differences in the points of view of stakeholders on the functional capabilities of object into account.

Use of the proposed models allows one to increase the degree of formalization of the functional requirements description on the early stages of designing, which makes it possible to develop formal conformity assessment procedures of properties of the base object to the desires and expectations of the non-homogeneous stakeholders.

Definitions of such concepts as "basic object", "reference object", "defect", "useful function", "harmful function" are given.

Offered is the procedure for assessing the significance of defects by set of estimates of stakeholders group, based on use of membership functions.

Keywords: *defect, defectology, stakeholder, membership function, hardware- software complex, harmful function, useful function*

Acknowledgements: *This work was supported by the Russian Foundation for Basic Research, project nos.16-08-00442.*

For citation:

Gvozdev V. E., Blinova D. V. Analysis of Functional Possibilities of Hardware-Software Complexes in the Early Stages of Designing with Considering the Opinions of Stakeholders, *Programmnaya Ingeneria*, 2016, vol. 7, no. 9, pp. 395—399.

DOI: 10.17587/prin.7.395-399

References

1. **Markov A. S., Fadin A. A.** Sistematika ujazvimostei i defektov bezopasnosti programmnih resursov (Taxonomy vulnerabilities and defects of software safety), *Zascshita informacii INSIDE,* 2013, vol. 3, pp. 2—7 (in Russian).

2. **Borodakij Yu. V., Yusupov R. M., Palchun B. P.** Problema imitacionnogo modelirovanija defectoskopicheskih svoistv kompyuternoj infosferi (Problem of imitation modelling of computing infosphere defectoscopy features), *Proceedings of the 3rd Russian conf. "Imitation modelling. Theory and Practice"*, Saint-Petersburg, 2007, pp. 87—92 (in Russian).

3. **Chai K. H., Zhang J., Tan K. C.** A TRIZ-Based Method for New Service Design, *Journal of Service Research*, 2005, vol. 8, issue 1, pp. 48—66.

4. **ISO/IEC** 15288:2002 System engineering — System life cycle processes.

5. **Lipaev V. V.** *Analiz i sokrashenie riskov proektov slozhnix programmnih sredstv* (Analysis and reduction of risks of complex software projects), Moscow, SINTEG, 2005, 224 p. (in Russian).

6. **Gvozdev V. E., Blinova D. V.** Ontologicheskii analiz defektov pri proektirovanii komponentov apparatno-programmnih kompleksov (Ontological analysis of defects at the designing of hardware-software complexes), *Ontology of Designing*, 2015, vol. 5, issue 4 (18), pp. 399—410 (in Russian).

7. **Lapigin Yu. N.** *Strategicheskij menedgment: Uchebnoe posobie* (Strategic Management: Textbook), Moscow, Vishee obrazovanie, 2007, 174 p. (in Russian).

8. **Gvozdev V. E., Blinova D. V.** Elementi teorii upravlenija neprednamerennimi defectami v informacionno-kommunikacionnih sistemax (Elements of the management theory inadvertent defects in information and communication systems), *Sistemnaja injenerija*, 2015, vol. 2, pp. 104—113 (in Russian).

9. **Chekmarev A. I.** *Kvalimetria i upravlenie kachestvom. Chast 1. Kvalimetria: uchebnoe posobie* (Qualimetry and quality management. Part 1. Qualimetry: Textbook), Samara, SSAU, 2010, 172 p. (in Russian).

10. **CHAOS** Manifesto 2011: The Laws of CHAOS and the CHAOS 100 Best PM Practices. The Standish Group International, Incorporated.

11. **Myers G. J.** *Software reliability: Principles and practices,* New York, Wiley, 1976. 360 p.

12. **Cherkesov G. N.** *Nadejnost' apparatno-programmnih kompleksov: uchebnoe posobie* (The reliability of hardware and software systems: Textbook). Saint-Petersburg, Piter, 2005, 479 p. (in Russian).

13. **Pelaez C. E., Bowles J. B.** Using fuzzy cognitive maps as a system model for failure modes and effects analysis, *Information Sciences*, 1996, vol. 88 (1), pp. 177—199.

14. **Kuznetsov O. P., Kulinich A. A., Markovskii A. V.** Analiz vlijanij pri upravlenii slabostrukturirovannimi situacijami na osnove kognitivnih kart (Analysis of the influence at management of semi-structured situations on the basis of cognitive maps), *Chelovecheskii factor v upravlenii*, Moscow, KomKniga, 2006, pp. 313—344 (in Russian)

УДК 004.7 DOI: 10.17587/prin.7.400-407

А. А. Андреев, магистрант, e-mail: andreev@cs.petrsu.ru, **А. С. Колосов**, ст. преподаватель, e-mail: akolosov@cs.petrsu.ru, **А. В. Воронин,** д-р техн. наук, ректор, e-mail: voronin@petrsu.ru, **Ю. А. Богоявленский,** канд. техн. наук, зав. каф., e-mail: ybgv@cs.petrsu.ru, Петрозаводский государственный университет

Обобщенная графовая модель структуры физического, канального и сетевого уровней ИКТ-инфраструктуры локального поставщика сетевых услуг

Предложена модель графа структуры базовых уровней ИКТ-инфраструктуры современных локальных поставщиков сетевых услуг, построенной на базе технологии Ethernet и протокола IP. Модель предоставляет возможность единообразно описывать различные алгоритмы сбора фактов о структуре сети, а также позволяет формально обосновать методы построения вершин и ребер графа, которые не могут быть построены на основе анализа данных, получаемых из сети.

Ключевые слова: *сетевое управление, поставщик сетевых услуг, граф ИКТ-инфраструктуры, обнаружение топологии, графовая модель*

Введение

Основные тенденции рынка информационных технологий [1], среди которых рост числа мобильных пользователей и широкое внедрение технологий виртуализации, обуславливают рост масштабов и сложности ИКТ-инфраструктур (далее — Сети) современных поставщиков сетевых услуг (ПСУ). Самым многочисленным классом ПСУ [2] являются предприятия средней величины, решающие ИТ-задачи на базе собственной Сети — локальные ПСУ (лПСУ).

Ключевым фактором обеспечения надежности и качества предоставляемых услуг является систематический подход к управлению Сетью ПСУ (методы сетевого управления), эффективная реализация которого требует наличия полного и детального описания структуры Сети. При этом для лПСУ, обслуживающих все три базовых уровня собственной Сети (физический, канальный и сетевой), важно описание структуры каждого из них. Так, например, широко применяемая в современных лПСУ технология виртуальных локальных вычислительных сетей (ВЛВС) существенно повышает гибкость организации физического уровня Сети за счет усложнения структуры канального уровня. Это, в конечном счете, сказывается на решении таких задач сетевого управления, как планирование мощности, поиск причин неисправностей, сопровождение сетевой документации и т. д.

Распространенной формой представления описания структуры Сети на заданном уровне является граф, вершины которого соответствуют управляемым элементам (сетевым устройствам, сетевым портам, конечным точкам протоколов передачи данных и т. д.), а ребра — связям иерархии и связям передачи данных. Одной из главных проблем автоматизации процесса построения графа структуры Сети является отсутствие стандартных средств обнаружения элементов сетевого окружения и связей между ними. Это обстоятельство приводит к необходимости анализа разнородных источников данных, не специализированных для этой задачи: таблиц коммутации, результатов работы протокола STP, кешей протоколов LLDP, CDP и ARP, таблиц маршрутизации протокола IP, и т. п. При этом ни одна из реализаций протоколов и технологий, предоставляющих подобные источники, не может гарантировать полноты и актуальности информации, необходимых для построения графа структуры Сети. На практике применение существующих методов обработки упомянутых выше источников осложняется отсутствием или недоступностью унифицированных интерфейсов доступа к ним. Таким образом, граф структуры Сети, наиболее близкий к актуальному, может быть получен только при обработке всех доступных источников данных о ее структуре. В таких условиях целесообразно [3, 4] разделить задачу построения графа структуры Сети на две части:

1) сбор фактов о текущем состоянии Сети из всех доступных источников;

2) анализ собранных фактов в целях построения фрагментов графа.

Данный подход приводит к необходимости построения модели, определяющей закономерности структур современных Сетей, на основе которой могут быть составлены графы разнообразных конкретных Сетей. Такую модель будем называть обобщенной графовой моделью структуры Сети.

Существующие алгоритмы основаны на использовании одного источника данных и отражают в графе только узкий набор структурных особенностей заданного уровня Сети. В связи с этим обстоятельством графовые модели, лежащие в основе этих алгоритмов, существенно ограничены и не могут быть использованы в качестве обобщенной модели структуры Сети. Так, в работах [5, 6] граф структуры Сети строится на основе доступных таблиц коммутации, поэтому их модели описывают только канальный уровень. Аналогично, в работе [7] Сеть рассматривается только с точки зрения работы служебного протокола канального уровня STP. Возможность наличия в Сети ВЛВС, построенных на основе стандарта IEEE 802.1Q, частично учитывается в работе [4], однако предложенная модель не позволяет описать организацию сетевого уровня. В модели алгоритма [8] учитывается наличие в Сети маршрутизаторов и IP-подсетей, однако ряд принятых авторами допущений не позволяет применять ее для описания сложных современных Сетей. Обобщенная графовая модель структуры физического, канального и сетевого уровней Сети лПСУ, рассматриваемая в настоящей статье, была разработана в рамках проекта Nest [9] в целях унификации описания алгоритмов сбора фактов о структуре Сети, а также формального обоснования методов построения фрагментов графа, данные о которых не могут быть получены от сетевых устройств.

Обобщенная графовая модель структуры Сети

Структура Сети в целом определяется совокупностью структур ее физического, канального и сетевого уровней, а также иерархическими связями между элементами этих структур. Наиболее распространенными технологиями реализации указанных уровней в Сетях лПСУ являются Ethernet и IP. Исходя из этого, сформулируем основные требования к разрабатываемой модели.

Модель должна предоставлять возможность для описания базовых объектов физического уровня Сети: устройство и порт сетевого подключения, а также выражать факты принадлежности порта устройству и физического соединения двух портов средой передачи данных. В модели должна учитываться возможность наличия в Сети агрегированных каналов, образованных путем транкинга портов в соответствии со стандартом IEEE 802.3ad. Взаимодействие со средой передачи данных посредством одного или более портов реализуется на уровне операционной системы устройства, которая, в свою очередь, предоставляет своим остальным компонентам независимую от реализации абстракцию точки доступа к среде передачи данных — физический интерфейс. Данная абстракция также должна быть отражена в модели.

Основной структурной единицей канального уровня, реализованного на основе Ethernet, является широковещательный домен — совокупность портов разных устройств, имеющих возможность передавать друг другу кадры данных. При этом, в случае применения ВЛВС стандарта IEEE 802.1Q, каждый порт может входить в несколько широковещательных доменов, идентифицируемых целочисленными метками. Модель должна позволять описывать любой вариант организации широковещательных доменов, а также отражать особенности работы коммутаторов, определенных стандартом IEEE 802.1D.

Структура сетевого уровня, реализованного на базе протокола IP, определяется логическим разбиением портов каждого широковещательного домена на IP-подсети. Каждый порт может относиться к нескольким подсетям. Модель должна позволять выражать группировки портов по подсетям, а также связи маршрутизации между подсетями. В рамках данной модели не рассматриваются случаи сложных политик маршрутизации, а также применение методов трансляции сетевых адресов (NAT).

По ходу изложения элементы модели будем иллюстрировать на примере демонстрационной Сети (рис. 1), состоящей из трех IP-подсетей, содержащих по одной рабочей станции. Каждая подсеть реализована в рамках отдельной ВЛВС. Разграничение широковещательных доменов между ВЛВС обеспечивается двумя коммутаторами, к которым физически подключены рабочие станции, а также интерфейсы маршрутизатора, обеспечивающего связь между подсетями. Соединения в рамках разных ВЛВС на рис. 1 обозначены разными видами линий.

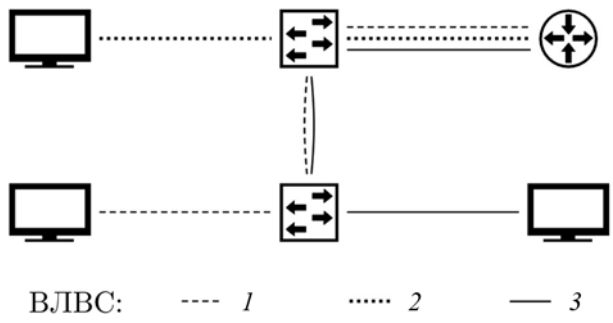


Рис. 1. Схема демонстрационной Сети

Физический уровень

Рассмотрим непустое конечное множество сетевых устройств D. Множество всех портов устройства $d \in D$ обозначим P_d, множество всех портов — P. Для любого $d \in D$ множество P_d непустое и конечное. Для любых двух $d_1, d_2 \in D$, $P_{d1} \cap P_{d2} = \varnothing$. Определим между портами и устройствами бинарное отношение ассоциации A^1 так, что $(p, d) \in A^1$ и $(d, p) \in A^1$ тогда

и только тогда, когда $p \in P$, $d \in D$ и $p \in P_d$. Отношение A^1 симметрично и антирефлексивно.

Рассмотрим на множестве P симметричное, транзитивное и антирефлексивное бинарное отношение L^1. Порты, находящиеся в отношении L^1, должны быть ассоциированы с разными устройствами, т. е. $\forall (p_1, p_2) \in L^1$ не существует $d \in D$, такого, что $p_1, p_2 \in P_d$. Дадим отношению L^1 следующую интерпретацию: порты $(p_1, p_2) \in L^1$ подключены к единой среде передачи данных.

Множество всех физических интерфейсов устройства $d \in D$ является разбиением множества его портов. Обозначим данное разбиение I^1_d, а множество всех физических интерфейсов всех устройств — I^1.

Структура физического уровня Сети может быть описана связным неориентированным графом $G^1 = \langle V^1, E^1 \rangle$, множество вершин которого содержит устройства и порты $(V^1 = D \cup P)$, а множество ребер — связи ассоциации между устройствами и портами, а также связи соединения на физическом уровне $(E^1 = A^1 \cup L^1)$. Граф структуры физического уровня демонстрационной Сети (см. рис. 1) представлен на рис. 2, где вершины, соответствующие устройствам, изображены квадратами, а вершины, соответствующие портам, — окружностями.

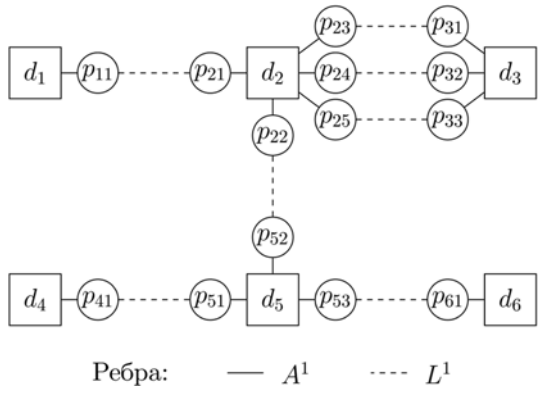


Рис. 2. Граф структуры физического уровня демонстрационной Сети

Канальный уровень

Назначим каждому физическому интерфейсу $pi \in I^1$ конечное множество меток $VID_{pi} \subset \mathbb{N}_0$, соответствующих идентификаторам ВЛВС, сконфигурированных на нем. В случае, если устройство, ассоциированное с портами из pi, не поддерживает технологию ВЛВС, $VID_{pi} \subset \{0\}$. Также поставим в соответствие каждому устройству $d \in D$ множество VID_d меток ВЛВС, в которые входят все его порты.

По аналогии с физическим уровнем, операционная система каждого устройства $d \in D$ создает для работы с ВЛВС, сконфигурированными на каждом из ее физических интерфейсов, отдельный канальный интерфейс (pi, υ), где $pi \in I^1_d$, $\upsilon \in VID_{pi}$. Непустое множество всех канальных интерфейсов всех устройств обозначим I^2. Определим между канальными интерфейсами и устройствами бинарное отношение ассоциации A^2 так, что $(li, d) \in A^2$ и $(d, li) \in A^2$ тогда и только тогда, когда $li = (pi, \upsilon) \in I^2$, $d \in D$ и для всех $p \in pi$ выполняется $(p, d) \in A^1$. Отношение A^2 симметрично и антирефлексивно. Множество всех канальных интерфейсов, ассоциированных с некоторым устройством d, будем обозначать I^2_d.

Определим на множестве I^2 бинарное отношение соединения на канальном уровне L^2 так, что два незаблокированных канальных интерфейса $li_1 = (pi_1, \upsilon_1) \in I^2_{d1}$, $li_2 = (pi_2, \upsilon_2) \in I^2_{d2}$, ассоциированные с разными устройствами, находятся в отношении L^2, если существуют $p_1 \in pi_1$, $p_2 \in pi_2$ такие, что $(p_1, p_2) \in L^1$. Отношение L^2 симметрично, транзитивно и антирефлексивно. Дадим отношению L^2 следующую интерпретацию: два устройства могут взаимодействовать на канальном уровне посредством незаблокированных (например, вследствие работы протокола STP) канальных интерфейсов, соединенных между собой на физическом уровне.

Некоторые устройства Сети (коммутаторы) могут быть сконфигурированы так, чтобы пересылать транзитные кадры между своими канальными интерфейсами. Рассмотрим на множестве I^2 симметричное, транзитивное и антирефлексивное бинарное отношение коммутации F^2. Интерфейсы, находящиеся в отношении F^2, должны быть ассоциированы с одним устройством, т. е. $\forall (li_1, li_2) \in F^2$ существует $d \in D$, такое, что $li_1, li_2 \in I^2_d$. Дадим отношению F^2 следующую интерпретацию: конфигурация устройства $d \in D$ предусматривает возможность пересылки кадров между двумя различными канальными интерфейсами $(li_1, li_2) \in F^2$.

Структура канального уровня Сети может быть описана связным неориентированным графом $G^2 = \langle V^2, E^2 \rangle$, множество вершин которого содержит устройства и канальные интерфейсы $(V^2 = D \cup I^2)$, а множество ребер — связи ассоциации между интерфейсами и устройствами, связи коммутации и соединения на канальном уровне $(E^2 = A^2 \cup F^2 \cup L^2)$. Граф структуры канального уровня демонстрационной Сети представлен на рис. 3, где вершины, соответствующие канальным интерфейсам, изображены эллипсами, внутри которых указаны порт (см. рис. 2)

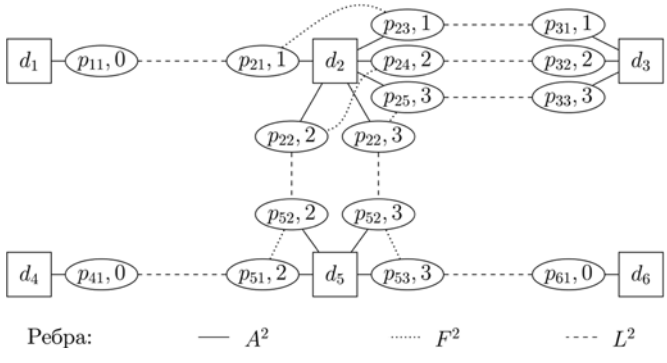


Рис. 3. Граф структуры канального уровня демонстрационной Сети

и идентификатор ВЛВС, определяющие данный интерфейс. Устройства изображены квадратами.

Рассмотрим граф $\widehat{G}^2 = \langle I^2, F^2 \cup L^2 \rangle$, являющийся подграфом G^2. Наличие пути между канальными интерфейсами в графе $\widehat{G}^2$ соответствует возможности их взаимодействия на канальном уровне либо напрямую, либо посредством цепочки коммутирующих устройств. Таким образом, множества вершин компонент связности графа $\widehat{G}^2$ являются широковещательными доменами Сети. Разбиение множества I^2, каждый элемент которого соответствует одному широковещательному домену, обозначим *BD*.

Назовем путем канального уровня такой реберно-простой путь в графе $\widehat{G}^2$, в котором нет двух подряд идущих ребер коммутации. Согласно стандарту IEEE 802.1D в графе G^2 между двумя любыми интерфейсами канального уровня не должно существовать более одного пути канального уровня.

Свойство 1. В графе G^2 между двумя интерфейсами канального уровня путь канального уровня существует тогда и только тогда, когда они входят в один широковещательный домен.

Доказательство. Тривиально, следует из определений широковещательного домена и пути канального уровня. ■

Сетевой уровень

Введем конечное множество N идентификаторов всех подсетей в Сети. Для каждой подсети с идентификатором $n \in N$ определено конечное множество возможных идентификаторов ее хостов — H_n. В целях передачи данных на сетевом уровне посредством одного или нескольких канальных интерфейсов операционная система каждого устройства $d \in D$ создает сетевой интерфейс (S, n, h), где $S \subset I_d^2$, $n \in N$, $h \in H_n$. Непустое множество всех сетевых интерфейсов всех устройств обозначим I^3. Определим между сетевыми интерфейсами и устройствами бинарное отношение ассоциации A^3 так, что $(ni, d) \in A^3$ и $(d, ni) \in A^3$ тогда и только тогда, когда $ni = (S, n, h) \in I^3$, $d \in D$ и для всех $li \in S$ выполняется $(li, d) \in A^2$. Отношение A^3 симметрично и антирефлексивно. Множество всех сетевых интерфейсов, ассоциированных с некоторым устройством d, будем обозначать I_d^3.

Два устройства могут взаимодействовать на сетевом уровне посредством сетевых интерфейсов, находящихся в одном широковещательном домене и имеющих одинаковый идентификатор подсети. Определим на множестве I^3 бинарное отношение соединения на сетевом уровне L^3 так, что два сетевых интерфейса $ni_1 = (S_1, n_1, h_1) \in I^3$ и $ni_2 = (S_2, n_2, h_2) \in I^3$, ассоциированные с разными устройствами, находятся в отношении L^3, если $n_1 = n_2$ и существуют $dom \in BD$, $li_1 \in S_1$, $li_2 \in S_2$ такие, что $li_1, li_2 \in dom.$. Отношение L^3 симметрично, транзитивно и антирефлексивно.

В каждой подсети могут присутствовать устройства (маршрутизаторы), способные осуществлять пересылку транзитных дейтаграмм между подключенными подсетями. По аналогии с канальным уровнем рассмотрим на множестве I^3 симметричное, транзитивное и антирефлексивное бинарное отношение маршрутизации F^3. Интерфейсы, состоящие в отношении F^3, должны быть ассоциированы с одним устройством, т. е. $\forall (ni_1, ni_2) \in F^3$ существует $d \in D$, такое, что $ni_1, ni_2 \in I_d^3$. Дадим отношению F^3 следующую интерпретацию: конфигурация устройства $d \in D$ предусматривает возможность пересылки дейтаграмм между двумя сетевыми интерфейсами $(ni_1, ni_2) \in F^3$.

Структура Сети на сетевом уровне может быть описана связным неориентированным графом $G^3 = \langle V^3, E^3 \rangle$, множество вершин которого содержит устройства и сетевые интерфейсы $(V^3 = D \cup I^3)$, а множество ребер — связи ассоциации между устройствами и сетевыми интерфейсами, связи маршрутизации и соединения на канальном уровне $(E^3 = A^3 \cup L^3 \cup F^3)$. Граф структуры сетевого уровня демонстрационной Сети представлен на рис. 4, где вершины, соответствующие сетевым интерфейсам, изображены прямоугольниками, внутри которых указаны канальный интерфейс, идентификатор подсети и идентификатор хоста, определяющие данный интерфейс. Здесь $li_{11} = (p_{11}, 0)$, $li_{31} = (p_{31}, 1)$, $li_{32} = (p_{32}, 2)$, $li_{33} = (p_{33}, 3)$, $li_{41} = (p_{41}, 0)$, $li_{61} = (p_{61}, 0)$. Устройства изображены квадратами.

Рассмотрим граф $\widehat{G}^3 = \langle I^3, L^3 \rangle$, являющийся подграфом G^3. Наличие пути между сетевыми интерфейсами в графе $\widehat{G}^3$ соответствует возможности их взаимодействия на сетевом уровне напрямую. Таким образом, множества вершин компонент связности графа $\widehat{G}^3$ являются подсетями Сети. Разбиение множества I^3, каждый элемент которого соответствует одной подсети, обозначим *SN*.

Назовем путем сетевого уровня такой реберно-простой путь в подграфе $\langle I^3, L^3 \cup F^3 \rangle$ графа G^3, в котором нет двух подряд идущих ребер маршрутизации или соединения на сетевом уровне.

Свойство 2. Путь сетевого уровня между двумя сетевыми интерфейсами содержит ребро маршрутизации тогда и только тогда, когда они принадлежат разным подсетям.

Доказательство. Тривиально, следует из определений подсети и пути сетевого уровня. ■

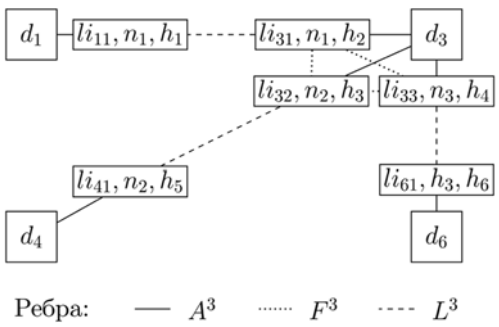


Рис. 4. Граф структуры сетевого уровня демонстрационной Сети

Структура любой заданной Сети может быть описана связным неориентированным графом $G = \langle V, E \rangle$ — графом структуры Сети, множество вершин которого $V = V^1 \cup V^2 \cup V^3 = D \cup P \cup I^2 \cup I^3$, а множество ребер $E = E^1 \cup E^2 \cup E^3 = A^1 \cup L^1 \cup A^2 \cup F^2 \cup L^2 \cup A^3 \cup F^3 \cup L^3$. Граф структуры Сети не содержит петель и кратных ребер, может содержать циклы.

Построение графа структуры Сети

Процесс построения графа структуры Сети по заданному набору данных, полученных в ходе опроса сетевых устройств (например, по протоколу SNMP), состоит из трех этапов. На первом этапе происходит построение фрагментов графа, описывающих устройства, существование которых напрямую следует из результатов анализа входных данных, совместно с их портами и интерфейсами.

При условии полноты входных данных, полученный на первом этапе граф описывает все устройства в Сети. Однако как уже было сказано во введении, данные условия не выполняются в реальных Сетях. Выявление устройств, сведения о которых отсутствуют во входных данных, выполняется в ходе второго этапа на основе анализа множеств достижимости канальных интерфейсов [5]. Так, например, на втором этапе происходит построение фрагментов графа, соответствующих неуправляемым концентраторам и коммутаторам.

Целью третьего этапа является построение ребер соединения канального уровня на основе анализа множеств достижимости, а также ребер соединения физического и сетевого уровней на основе определений модели.

Описанный процесс построения графа структуры Сети был реализован в рамках подсистемы Nestopo экспериментальной платформы Nest [10]. В качестве источников входных данных в Nestopo используются MIB сетевых устройств, доступ к которым осуществляется по протоколу SNMP.

Рассмотрим подробнее методы построения и анализа множеств достижимости канальных интерфейсов, применяемые на втором и третьем этапах.

Методы построения множеств достижимости канальных интерфейсов

В Сети один интерфейс канального уровня достижим от другого, если может являться для него конечной точкой передачи данных на канальном уровне. Данные о достижимости между интерфейсами позволяют выявить наличие сетевых устройств, сведения о которых отсутствуют во входных данных процесса построения графа, и ребер соединения на канальном уровне.

Рассмотрим понятие достижимости в рамках графа структуры Сети *G*. Назовем путем достижимости такой путь канального уровня, первое и последнее ребра в котором не являются ребрами коммутации. Введем на множестве I^2 бинарное симметричное антирефлексивное отношение достижимости $\leftrightarrow$ такое, что $li_1 \leftrightarrow li_2$, если от li_1 до li_2 существует путь достижимости в графе *G*. В этом случае будем говорить, что li_2 достижим от li_1. В графе структуры канального уровня демонстрационной Сети (см. рис. 3) в широковещательном домене, образованным ВЛВС с идентификатором 1, в отношении достижимости состоят пары интерфейсов $(\{p_{11}\}, 0)$ и $(\{p_{21}\}, 1)$, $(\{p_{23}\}, 1)$ и $(\{p_{31}\}, 1)$, $(\{p_{11}\}, 0)$ и $(\{p_{31}\}, 1)$.

Будем говорить, что порт $p \in P$ достижим от интерфейса $li_1 \in I^2$, если существует интерфейс $(pi, v) \in I^2$, достижимый от li_1, для которого $p \in pi$.

Для каждого $li \in I^2$ введем множество $RS_{li} \subset I^2$, содержащее все интерфейсы, достижимые от *li*. Множество интерфейсов, достижимых от состоящих в отношении коммутации с *li* интерфейсов, будем обозначать CRS_{li}. В графе на рис. 3 множество *RS* интерфейса $(\{p_{22}\}, 3)$ равно $\{(\{p_{52}\}, 3), (\{p_{61}\}, 0)\}$, а множество *CRS* равно $\{(\{p_{33}\}, 3)\}$.

Множества достижимости могут быть построены по входным данным процесса построения графа структуры Сети. Большинство источников данных предоставляют информацию о достижимости портов от интерфейсов. По определению достижимости и свойству 1 для достижимости порта *p* от интерфейса li_1 необходимо вхождение интерфейса $li_1 = (pi_1, v_1)$ и какого-либо $li_2 = (pi_2, v_2)$ (где $p \in pi_2$) в один широковещательный домен. При отсутствии данных о вхождении достижимого порта в ВЛВС, будем считать, что либо $v_1 = v_2$ (если известно, что устройство поддерживает ВЛВС), либо $v_2 = 0$.

Точность обнаружения элементов Сети путем анализа множеств достижимости напрямую зависит от полноты данных о достижимости [5, 11]. Проблема полноты и доступности данных о структуре Сети приводит к необходимости поиска косвенных данных о достижимости, что возможно с помощью следующего утверждения.

Утверждение 1. Если канальный интерфейс li_2 достижим от интерфейса li_1, то от li_1 достижимы все интерфейсы, достижимые от интерфейсов, коммутирующих с li_2, т. е. $CRS_{li2} \subset RS_{li1}$.

Доказательство. Пусть $li_1 \in I^2_{d1}$, $li_2, li_3 \in I^2_{d2}$, где $d_1, d_2 \in D$. Пусть при этом $(li_2, li_3) \in F^2$ и $li_1 \leftrightarrow li_2$, а также существует интерфейс $li_4 \in I^2_{d3}$ (где $d_3 \in D$) такой, что $li_3 \leftrightarrow li_4$. По определению достижимости между интерфейсами существуют пути достижимости $(li_1, ..., li_2)$ и $(li_3, ..., li_4)$. Но, так как $(li_2, li_3) \in F^2$, значит в графе *G* существует путь $(li_1, ..., li_2, li_3, ..., li_4)$, удовлетворяющий определению пути достижимости. Значит $li_4 \in RS_{li1}$. ■

Построение множеств достижимости проводится в начале второго этапа построения графа структуры Сети, после чего они дополняются с помощью свойства симметричности отношения достижимости и утверждения 1.

Методы выявления устройств

Назовем устройство, два и более интерфейсов которого находятся в отношении соединения с интерфейсами других устройств, промежуточными для этих устройств. Наличие промежуточного устройства может быть установлено в ходе анализа множеств достижимости с помощью следующего утверждения.

Утверждение 2. Если два интерфейса li_1 и li_2, достижимые друг от друга, имеют общие достижимые интерфейсы, то эти общие интерфейсы не ассоциированы с устройствами, интерфейсы которых находятся на пути достижимости между li_1 и li_2.

Доказательство. Рассмотрим интерфейсы $li_1, li_2, li_3 \in I^2$. Пусть li_3 достижим от li_1 и li_2, а $li_1 \leftrightarrow li_2$, т. е. существует единственный путь достижимости $w = (li_1, ..., li_2)$. Допустим, что $w = (li_1, ..., li_3, ..., li_2)$. Обозначим связанные с li_3 интерфейсы, которые стоят на пути w ближе и дальше li_3, как li_4 и li_5 соответственно (при этом li_4 может совпадать с li_1, а li_5 — с li_2). По определению достижимости li_3 должен состоять с li_4 и li_5 либо в отношении L^2, либо F^2. По определению отношения L^2 интерфейс li_3 может состоять в нем только с одним другим интерфейсом, значит, по определению достижимости, li_3 не может одновременно быть достижимым и от li_1 и от li_2, т. е. li_3 не лежит на пути w. При этом li_3 не может состоять в отношении F^2 ни с li_4, ни с li_3 (и, соответственно, быть ассоциирован с устройствами, ассоциированными с li_4 и li_5), так как путь от li_1 или li_2 до li_3 будет оканчиваться ребром коммутации, что противоречит условию достижимости li_3 от li_1 и li_2. ■

Принадлежность интерфейсов к промежуточному устройству может быть установлена с помощью следующего утверждения.

Утверждение 3. Если для двух достижимых друг от друга канальных интерфейсов li_1 и li_2 выполняются следующие условия: 1) существует интерфейс li_3, достижимый от одного из двух интерфейсов и не достижимый от другого; 2) li_3 не достижим от интерфейсов, коммутирующих с li_1 или li_2; тогда li_3 лежит на пути достижимости между li_1 и li_2.

Доказательство. Пусть $li_1 \leftrightarrow li_2$, и интерфейс li_3 достижим от li_1, но не достижим от li_2 и интерфейсов в отношении F^2 с ним. Интерфейсы li_1, li_2, li_3 находятся в одном широковещательном домене $dom \in BD$ (по свойству 1). Так как $li_1 \leftrightarrow li_3$, то путь канального уровня $w_1 = (li_1, ..., li_3)$ существует и не имеет с концов ребер коммутации. Путь канального уровня $w_2 = (li_3, ..., li_2)$ существует по свойству 1. Значит, по определению пути канального уровня и условию единственности таких путей, существует либо путь канального уровня $w_3 = (li_1, ..., li_3, ..., li_2)$, либо путь канального уровня $w_4 = (li_1, ..., li_2, ..., li_3)$. Путь w_4 не может быть путем канального уровня, так как он заканчивается ребром L^2 и по определению достижимости либо $li_2 \leftrightarrow li_3$, либо $li_3 \in CRS_{li2}$, что противоречит условию. Допустим, что путь w_3 не является путем достижимости. Значит, так как $li_1 \leftrightarrow li_2$, то существует путь канального уровня $w_5 = (li_1, ..., li_2) \neq w_3$, что противоречит условию единственности путей канального уровня. То есть li_3 находится на пути достижимости между li_1 и li_2. Аналогично доказательство для интерфейсов, достижимых от li_2 и не достижимых от li_1. ■

Следующее утверждение позволяет установить наличие ребра коммутации между интерфейсами промежуточного устройства.

Утверждение 4. Если для устройства $d \in D$ и двух достижимых друг от друга интерфейсов канального уровня li_1 и li_2 выполняются следующие условия: 1) от li_1 достижим $li_3 \in I_d^2$, от li_2 достижим $li_4 \in I_d^2$; 2) от li_1 и li_2 не достижимы интерфейсы из I_d^2, отличные от li_3 и li_4 соответственно; 3) от интерфейсов, коммутирующих с li_1 или li_2, не достижимы интерфейсы, ассоциированные с d; тогда li_3 и li_4 находятся в отношении F^2.

Доказательство. По утверждению 3 интерфейсы li_3 и li_4 лежат на пути достижимости $w = (li_1, ..., li_3, ..., li_4, ..., li_2)$. При этом подпути $w_1 = (li_1, ..., li_3)$ и $w_2 = (li_4, ..., li_2)$ пути w имеют с обоих концов ребра соединения канального уровня (отношение L^2) по определению достижимости. Так как li_3 и li_4 ассоциированы с одним устройством d, то они не могут находиться в отношении L^2 друг с другом. Допустим, что они не находятся в отношении F^2. Тогда подпуть $w_3 = (li_3, ..., li_4)$ пути w должен проходить через еще один интерфейс $li_5 \in I_d^2$, так как li_3 и li_4 уже фигурируют в ребрах соединения канального уровня в подпутях w_2 и w_3, а в отношении F^2 могут состоять только с интерфейсами из I_d^2. Для определенности положим, что $(li_3, li_5) \in F^2$. Тогда, по определению пути канального уровня, подпуть $w_4 = (li_5, ..., li_2)$ пути w не может начинаться с ребра коммутации. Но в таком случае путь w_4 удовлетворяет определению пути достижимости и $li_5 \leftrightarrow li_2$, что противоречит условию утверждения. Таким образом, $(li_3, li_4) \in F^2$. ■

Построение выявленных методом анализа множеств достижимости промежуточных устройств проводится на втором этапе процесса построения графа структуры Сети.

Методы построения ребер соединения

Выведем ряд критериев наличия ребра соединения между парами интерфейсов.

Первая теорема дает критерий наличия связей с пограничными устройствами, каковыми могут являться рабочие станции или маршрутизаторы.

Теорема 1. Если от канального интерфейса li_1 достижим только один канальный интерфейс li_2, то li_1 и li_2 находятся в отношении L^2.

Доказательство. Так как $li_1 \leftrightarrow li_2$, значит в графе структуры Сети существует путь достижимости $w = (li_1, ..., li_2)$. Путь w не может включать другие интерфейсы устройств d_1 и d_2 по определению. Допустим, что w включает интерфейс li_3 какого-либо устройства d_3. Если $w = (li_1, li_3, ..., li_2)$, то из определения пути достижимости следует $(li_1, li_3) \in L^2$, а значит $li_1 \leftrightarrow li_3$, что противоречит условию теоремы. Если же $w = (li_1, ..., li_3, li_2)$, то из определения пути достижимости следует необходимость существования интерфейса $li_4 \in I_{d2}^2$ такого, что $(li_4, li_3) \in F^2$ и li_4 стоит в пути перед li_3: $w = (li_1, ..., li_4, li_3, li_2)$. Но тогда по определению достижимости $li_1 \leftrightarrow li_4$, что тоже противоречит условию. Таким образом, w не может включать других интерфейсов устройств d_1 и d_2 и интерфейсов других устройств, а значит $w = (li_1, li_2)$ и значит $(li_1, li_2) \in L^2$. ■

Вторая теорема дает критерий наличия связи между интерфейсами различных устройств.

Теорема 2. Интерфейсы li_1 и li_2 находятся в отношении L^2 тогда и только тогда, когда $RS_{li1} = CRS_{li2} \cup \{li_2\}$ и $RS_{li2} = CRS_{li1} \cup \{li_1\}$.

Доказательство. Пусть интерфейсы $li_1, li_2 \in I^2$ находятся в отношении L^2, тогда, по определению достижимости $li_2 \in RS_{li1}$ и, по утверждению 1, $CRS_{li2} \subset RS_{li1}$. По определению достижимости и условию единственности пути достижимости RS_{li1} не может включать другие канальные интерфейсы, так как любой путь до них проходил бы через интерфейс li_2 и коммутирующий с ним. Таким образом, $RS_{li1} = CRS_{li2} \cup \{li_2\}$. Аналогично для RS_{li2}.

Обратно, пусть $RS_{li1} = CRS_{li2} \cup \{li_2\}$. Допустим, что li_1 находится в отношении L^2 не с li_2, а с $li_3 \in CRS_{li2}$. Тогда, по определению достижимости, существует путь канального уровня $(li_1, ..., li_2)$, который не заканчивается ребром коммутации, и путь $(li_3, ..., li_2)$, который заканчивается ребром коммутации. Так как $(li_1, li_3) \in L^2$, то путь $(li_1, li_3, ..., li_2)$ тоже путь канального уровня, что невозможно в связи с ограничением модели на число путей. Аналогично для RS_{li2}. ■

Таким образом, на основе приведенных теорем и определений модели можно определить следующие критерии наличия ребер соединения.

Критерий 1. По теореме 1, если от интерфейса $li_1 \in I^2$ достижим только один интерфейс $li_2 \in I^2$, то $(li_1, li_2) \in L^2$.

Критерий 2. По теореме 2, если два интерфейса li_1 и li_2 достижимы друг от друга, и $RS_{li1} = CRS_{li2} \cup \{li_2\}$, то $(li_1, li_2) \in L^2$.

Критерий 3. По определению отношения L^2, если два канальных интерфейса $li_1 = (pi_1, \upsilon_1)$ и $li_2 = (pi_2, \upsilon_2)$ находятся в отношении L^2, то для каждого $p_1 \in pi_1$, существует $p_2 \in pi_2$ такой, что $(p_1, p_2) \in L^1$, и наоборот.

Критерий 4. По определению отношения L^2, если два порта p_1 и p_2 находятся в отношении L^1, существуют такие идентификатор ВЛВС $\upsilon \in VID_{p1} \cap VID_{p2}$ и физические интерфейсы pi_1, pi_2, что $p_1 \in pi_1$, $p_2 \in pi_2$, а также канальные интерфейсы $li_1 = (pi_1, \upsilon)$, $li_2 = (pi_2, \upsilon)$ при этом незаблокированы, то $(li_1, li_2) \in L^2$.

Критерий 5. По определению отношения L^3, если для двух сетевых интерфейсов $ni_1 = (S_1, n, h_1)$ и $ni_2 = (S_2, n, h_2)$ существуют интерфейсы $li_1 \in S_1$ и $li_2 \in S_2$, которые состоят в одном широковещательном домене, то $(ni_1, ni_2) \in L^3$.

Построение ребер графа с помощью перечисленных критериев выполняется на третьем этапе процесса построения графа структуры Сети.

Заключение

Современный подход к построению детального и полного графа структуры Сети лПСУ заключается в последовательном применении большого числа независимых методов получения и анализа разнородных источников данных о сетевых устройствах и связях между ними. Объединенная реализация данных методов в целях построения максимально точного и детального графа структуры Сети в рамках проекта Nest [9] потребовала разработки обобщенной графовой модели, позволяющей универсальным способом описывать графы разнообразных конкретных Сетей.

В статье предложена обобщенная графовая модель структуры физического, канального и сетевого уровней современной Сети лПСУ, построенной на базе технологии Ethernet и протокола IP. Описание модели содержит определения графов структуры каждого из уровней Сети и их свойства. На основе модели сформулирован ряд критериев для построения элементов графа Сети, не представленных явным образом в доступных данных о ее структуре. В статье также описан независимый от конкретных источников данных процесс построения графа структуры заданной Сети и его формальное обоснование с помощью предложенной модели.

В будущем планируется разработка дополнений к предложенной модели, позволяющих отражать в графе структуры Сети тоннели виртуальных частных сетей, виртуальные машины и сетевое оборудование, а также учитывающих особенности беспроводных сетевых технологий.

Список литературы

1. **Агапов В., Пратусевич В., Яковлев С.** Обзор и оценка перспектив развития мирового и российского рынков информационных технологий: Аналитический отчет. International Data Corporation (IDC), 2014. URL: http://fs.moex.com/files/9216.
2. **Sung Y.-W. E., Sun X., Rao S. G., Xie G. G., Maltz D. A.** Towards systematic design of enterprise networks // Networking, IEEE/ACM Transactions on. 2011. Vol. 19, N 3. P. 695—708.
3. **Воеводин В. В., Стефанов К. С.** Автоматическое определение и описание сетевой инфраструктуры суперкомпьютеров // Вычислительные методы и программирование. 2014. Т. 15, № 4. С. 560—568.
4. **Zichao L., Ziwei H., Geng Z., Yan M.** Ethernet topology discovery for virtual local area networks with incomplete information // Network infrastructure and digital content (IC-NIDC), 2014 4th IEEE International Conference on. 2014. P. 252—256.
5. **Gobjuka H., Breitbart Y.** Ethernet topology discovery for networks with incomplete information // Networking, IEEE/ACM Transactions on. 2010. Vol. 18, N 4. P. 1220—1233.
6. **Sun Y., Shi Z., Wu Z.** A discovery algorithm for physical topology in switched ethernets // Local computer networks, 2005. 30th anniversary The IEEE COnference on. IEEE, 2005. P. 311—317.
7. **Son M.-H., Joo B.-S., Kim B.-C., Lee J.-Y.** Physical topology discovery for metro ethernet networks // ETRI Journal. 2005. Vol. 4, N 27. P. 355—366.
8. **Bejerano Y.** Taking the skeletons out of the closets: A simple and efficient topology discovery scheme for large ethernet lans // Networking, IEEE/ACM Transactions on. 2009. Vol. 17, N 5. P. 1385—1398.
9. **Богоявленский Ю. А.** Прототип экспериментальной платформы Nest для исследования моделей и методов управления ИКТ-инфраструктурами локальных поставщиков услуг Интернет // Программная инженерия. 2013. № 2. С. 11—20.
10. **Андреев А. А., Колосов А. С., Богоявленский Ю. А.** Автоматизация построения графа канального уровня ИКТ-инфраструктуры локального поставщика услуг интернета // Ученые записки Петрозаводского государственного университета. Серия: Естественные и технические науки. Петрозаводск, 2015. № 2 (147). С. 97—102.
11. **Breitbart Y., Gobjuka H.** Characterization of layer-2 unique topologies // Information Processing Letters. 2008. Vol. 105, N 2. P. 52—57.

Generalized Graph Model of the Physical, Link and Network Layers Structure of the ICT-Infrastructure of a Local Network Service Provider

A. A. Andreev, andreev@cs.petrsu.ru, **A. S. Kolosov,** akolosov@cs.petrsu.ru,
A. V. Voronin, voronin@petrsu.ru, **Iu. A. Bogoiavlenskii,** ybgv@cs.petrsu.ru,
Petrozavodsk State University, 185910, Petrozavodsk, Russian Federation,

Corresponding author:

Andreev Anton A., Master Student, Petrozavodsk State University, 185910, Petrozavodsk, Russian Federation, e-mail: andreev@cs.karelia.ru

Received on June 23, 2016
Accepted on June 30, 2016

A systematic approach to management of a local network service provider's ICT-infrastructure (network) requires a topology graph of the network's three basic layers. Network topology discovery process is complicated due to lack of standard tools for devices detection, data incompleteness and heterogeneity. In such circumstances it is expedient to divide the problem of graph discovery into two parts: 1) collection of facts on the current network state from all available data sources; 2) analysis of collected facts in order to build graph fragments. This approach requires a generalized graph model of a Network structure, which could define a family of topology graphs of modern Networks.

This paper proposes a generalized graph model of physical, link and network layers topology of a modern local network service provider's Ethernet and IP based network. The model defines network topology graphs for each layer of an ICT-infrastructure and properties of this graphs. The paper also describes the process of building a network topology graph that is independent of specific data sources, and the formal rationale of its stages.

The proposed model and the Network graph building process allow one to develop an efficient software system for automatized Network topology discovery.

Keywords: *network management, network service provider, ICT-infrastructure graph, topology discovery, graph model*

For citation:

Andreev A. A., Kolosov A. S., Voronin A. V., Bogoiavlenskii Iu. A. Generalized Graph Model of the Physical, Link and Network Layers Structure of the ICT-infrastructure of a Local Network Service Provider, *Programmnaya Ingeneria*, 2016, vol. 7, no. 9, pp. 400—407.

DOI: 10.17587/prin.7.400-407

References

1. **Agapov V., Pratusevich V., Jakovlev S.** Obzor i ocenka perspektiv razvitija mirovogo i rossijskogo rynkov informacionnyh tehnologij (Review and assessment of prospects of development of world and Russian market of information technologies): Analytical report. International Data Corporation (IDC), 2014, available at: http://fs.moex.com/files/9216 (in Russian).

2. **Sung Y.-W. E., Sun X., Rao S. G., Xie G. G., Maltz D. A.** Towards systematic design of enterprise networks. *Networking, IEEE/ACM Transactions on,* IEEE, 2011, vol. 19, no. 3, pp. 695—708.

3. **Voevodin V. V., Stefanov K. S.** Avtomaticheskoe opredelenie i opisanie setevoj infrastruktury superkomp'juterov (Automatic Detection and Description of Supercomputer Network Infrastructure), *Vychislitel'nye metody i programmirovanie*, 2014, vol. 15, no. 4, pp. 560—568 (in Russian).

4. **Zichao L., Ziwei H., Geng Z., Yan M.** Ethernet topology discovery for virtual local area networks with incomplete information, *Network infrastructure and digital content (IC-NIDC),* 2014 4th IEEE International Conference on, 2014, pp. 252—256.

5. **Gobjuka H., Breitbart Y.** Ethernet topology discovery for networks with incomplete information, *Networking, IEEE/ACM Transactions on,* 2010, vol. 18, no. 4, pp. 1220—1233.

6. **Sun Y., Shi Z., Wu Z.** A discovery algorithm for physical topology in switched ethernets, *Local computer networks,* 2005, 30th anniversary the IEEE Conference on, IEEE, 2005, pp. 311—317.

7. **Son M.-H., Joo B.-S., Kim B.-C., Lee J.-Y.** Physical topology discovery for metro ethernet networks, *ETRI Journal,* 2005, vol. 4, no. 27, pp. 355—366.

8. **Bejerano Y.** Taking the skeletons out of the closets: A simple and efficient topology discovery scheme for large ethernet lans, *Networking, IEEE/ACM Transactions on*, 2009, vol. 17, no. 5, pp. 1385—1398.

9. **Bogoiavlenskii Iu. A.** Prototip jeksperimental'noj platformy Nest dlja issledovanija modelej i metodov upravlenija IKT-infrastrukturami lokal'nyh postavshhikov uslug Internet (Prototype of the Tested Nest for Research of Network Management Methods and Models at the Enterprise Network Level), *Programmnaya Ingeneria,* 2013, no. 2, pp. 11—20 (in Russian).

10. **Andreev A. A., Kolosov A. S., Bogoiavlenskii Iu. A.** Avtomatizacija postroenija grafa kanal'nogo urovnja IKT-infrastruktury lokal'nogo postavshhika uslug interneta (Automation of ICT-infrastructure link layer graph discovery for local internet service providers), *Uchenye zapiski Petrozavodskogo gosudarstvennogo universiteta. Serija: Estestvennye i tehnicheskie nauki*, 2015, no 2 (147), pp. 97—102 (in Russian).

11. **Breitbart Y., Gobjuka H.** Characterization of layer-2 unique topologies, *Information Processing Letters,* 2008, vol. 105, no. 2, pp. 52—57.

УДК 004.82 DOI: 10.17587/prin.7.408-413

С. А. Афонин, канд. физ.-мат. наук, вед. науч. сотр., e-mail: serg@msu.ru,
А. С. Козицын, канд. физ.-мат. наук, вед. науч. сотр., e-mail: alexanderkz@mail.ru,
Д. А. Шачнев, аспирант, e-mail: mitya57@mitya57.me, Московский государственный университет имени М. В. Ломоносова

Программные механизмы агрегации данных, основанные на онтологическом представлении структуры реляционной базы наукометрических данных

Описаны математическая модель и программная реализация агрегатора наукометрических данных, представляющих результаты деятельности отдельных субъектов (персон). Модель основана на использовании онтологической структуры категорий данных и на суммировании значений различных параметров (индикаторов), определяющих отдельные составляющие оценки деятельности, для каждой из этих категорий. На основе формул расчета строятся запросы к реляционной базе данных, на основе которых заполняется таблица с соответствующими формуле результатами деятельности. Модель может быть использована для поиска авторов или коллективов, показавших наиболее эффективные в какой-либо области результаты, а также для сбора статистики о деятельности подразделений. В качестве примера структуры базы данных использована структура информационно-аналитической системы "ИСТИНА" (Информационной Системы Тематического Исследования НАукометрических данных), которая разработана и в течение трех лет успешно эксплуатируется в МГУ имени М. В. Ломоносова. Механизмы агрегации данных прошли успешные испытания на этой системе.

Ключевые слова: *онтологии, семантическая паутина, наукометрические данные, генерация SQL-запросов, агрегация данных*

Актуальность задачи

В качестве систем на основе наукометрии за рубежом, как правило, используют Current Research Information Systems (CRIS-системы). К крупнейшим из их числа относятся, например, Elsevier Pure [1] и Thomson Reuters Converis [2]. Такие системы, как правило, основаны на учете и индексировании англоязычных публикаций.

Подобные системы существуют и в России. В качестве примеров можно упомянуть Электронный научный архив УрФУ (Екатеринбург) [3], Информационно-аналитическую систему (ИАС) Астраханского государственного университета [4] или ИАС сопровождения научно-исследовательской деятельности Санкт-Петербургского государственного университета [5].

Во всех перечисленных системах в качестве показателей эффективности деятельности подлежащего оценке субъекта обычно используют число цитирований статей или индекс Хирша. Некоторые организации позволяют суммировать импакт-факторы журналов, в которых были опубликованы статьи (в зарубежных системах это импакт-факторы Scopus и Web of Science, в отечественных к ним добавляется Российский индекс научного цитирования). Принципиальным отличием способа, описанного в настоящей статье, является возможность комбинировать *любые* имеющиеся в ИАС системе количественные показатели с помощью формулы, определяющей возможные классы результатов деятельности, весовые коэффициенты для этих классов и ограничения на параметры подлежащих учету результатов деятельности.

Общая структура онтологии

Информационно-аналитические системы имеют дело с большим количеством данных, в рассматриваемом случае — наукометрических данных. Одной из естественных логических моделей представления этих данных являются онтологии [6]. Основная онтологическая связь, представляющая интерес в контексте

настоящей статьи — это отношение *authorOf*, субъектом которой является автор, а объектом — результат деятельности этого автора, например, опубликованная статья или доклад на конференции. Похожим отношением является *memberOf*, задающее членство человека (субъекта оценки) в некотором коллективе, например, редколлегии журнала или оргкомитете конференции. Разумеется, каждый объект может быть связан такими отношениями с несколькими субъектами, например у статьи может быть несколько авторов. У некоторых объектов есть родительские объекты, например, для статьи это журнал или сборник, в котором она опубликована, а для доклада на конференции это сама конференция. Дочерние и родительские объекты соединены между собой отношениями *parentOf*. Физически каждому типу связи автора и объекта соответствует одна таблица в реляционной базе данных, например, для авторства статей — это таблица `AUTHORA`, а для авторства книг — таблица `AUTHORB`.

На рисунке представлен пример множества объектов и связей между ними.

Каждый объект относится к своему классу. Все классы организованы в соответствующее их отношениям дерево классов. Некоторые классы, как правило, верхние, соответствуют отдельным таблицам в реляционной базе данных, а их подклассы определяются на основе правил, например, верхним классом может являться класс "книга", а на основе проставленных пометок конкретная книга может принадлежать к классам "монография" или "учебник". Возможно и обратное, а именно объединение некоторых физических классов в один класс онтологии.

У объекта могут также быть некоторые свойства, тип которых, как правило, либо действительные числа, либо булевы значения. Свойства могут как храниться непосредственно в таблице, содержащей сам объект, так и вычисляться на основе более сложных запросов (например, наличие или число связей с другими объектами в базе).

Постановка задачи

Пусть даны несколько наборов объектов $\left\{O^k = O_1^k, \ldots, O_m^k\right\}$. Например, k-м набором может являться множество результатов деятельности (далее — работ) некоторого автора. Требуется создать структуру, позволяющую получить для каждого набора некоторый числовой показатель, учитывающий количества объектов в наборе, их свойства и заданные весовые коэффициенты для каждого класса объектов. Такую структуру будем называть формулой расчета.

Структура формулы расчета

Формула состоит из множества строк, каждая из которых определяет некоторое множество объектов и задает функцию, ставящую подходящему объекту в соответствие некоторое действительное число.

Каждая строка формулы представляет собой набор $L = (K^L, R^L, P^L, M^L, \omega^L)$, где

- K^L — класс онтологии;
- R^L — массив ограничений (каждое ограничение R_i^L это функция, ставящая в соответствие каждому объекту булево значение: 0 или 1);
- P^L — функция-свойство, ставит в соответствие объекту значение некоторого параметра, являющееся действительным числом;
- M^L — массив модификаторов, функций, принимающих на вход объект и старое значение, и возвращающих новое значение (примеры модификаторов: "разделить на число соавторов" или "ограничить сверху значением N");
- ω^L — весовой коэффициент, действительное число.

Определение 1. Будем считать, что строка L *подходит* для объекта O, если O является экземпляром класса K^L, и для каждого ограничения R_i^L выполнено $R_i^L(O) = 1$.

Определение 2. *Функция аппликации* $\alpha(L, O)$ определена как $\alpha(L,O) = \left(M_1^L \circ \cdots \circ M_q^L\right)\left(O, \omega^L \cdot P^L(O)\right)$, если L подходит для O, и $\alpha(L, O) = 0$ иначе.

Пусть теперь $L_1, \ldots, L_n$ — строки, $O_1, \ldots, O_m$ — объекты. При подсчете для каждого из объектов должна выбираться строка, для которой функция аппликации возвращает наибольшее значение.

Определение 3. *Результатом выполнения формулы* называется число

$$S = \sum_{j=1}^{m} \max_{1 \leqslant i \leqslant n} \alpha\left(L_i, O_j\right).$$

Машинным представлением формулы является массив в формате JSON (JavaScript Object Notation). Каждым элементом массива является JSON-объект со следующими атрибутами:

- *category* — название класса K^L онтологии в формате *корневой_класс.класс1...классN*, напри-

мер *articles.journals.international* — от самого общего, корневого, к самому узкому классу;

- *label* — заголовок, демонстрируемый человеку, на русском языке (при создании формулы по умолчанию генерируется на основе названия категории);
- *restrictions* — массив с ограничениями R^L; каждое ограничение является объектом и содержит название параметра *parameter*, для числовых ограничений — значения верхней (*upper_bound*) и/или нижней (*lower_bound*) границы допустимых значений параметра, для булевых ограничений — их требуемое значение *bool_value;*
- *parameter* — название свойства P^L, которое будет учитываться в системе. Для удобства некоторые названия свойств могут быть сгруппированы в древовидную структуру как названия категорий, однако алгоритмы эту группировку никак не учитывают, в отличие от категорий;
- *weight* — значение весового коэффициента ω^L;
- *modifiers* — массив модификаторов M^L.

Каждый из возможных модификаторов задан отдельным классом на языке JavaScript, который предоставляет название, графический интерфейс для ввода данных и их текстовое представление. В формуле хранится название класса и массив его параметров (длина массива зависит от класса). Например, *limit_sum undefined 10* означает "ограничить суммарный балл по строке, нижнего ограничения нет, верхнее ограничение 10 баллов".

Отдельными атрибутами задаются ограничения, связанные со ссылками на другие классы онтологии, если таковые имеются. Например, если от журнала требуется принадлежность списку журналов с идентификатором ID, то добавляется атрибут *journalList: ID;* если для учебного курса требуется, чтобы вид обучения имел код *N*, то добавляется атрибут *educStage: N.*

Помимо массива JSON, в формуле также хранятся ее название, год начала и год окончания подсчета. Отдельно доступна история изменения формулы. Формула связана также отношениями с пользователем, создавшим ее, и с одним или несколькими подразделениями. Последний вид отношений используется как в интерфейсе, так и при подсчете: для некоторых классов работ есть возможность учитывать только те работы, которые имеют связь с тем же подразделением, что и формула.

Привязки параметров к полям в базе данных бывают нескольких типов, перечисленных далее.

- Прямое соответствие полю в основной таблице. Например, параметр "число работ" для категории "педагогическая работа" соответствует полю `F_EDUCWORK_WORKCOUNT` в таблице `EDUCWORK`.
- Соответствие комбинации значений полей, например, сумме или произведению. В той же категории параметр "общее число студентов" равен сумме полей `F_EDUCWORK_STUDBUDGET` и `F_EDUCWORK_STUDPAID`, а параметр "число часов в курсе" равен произведению полей `F_EDUCWORK_WEEK` и `F_EDUCWORK_HOUR`.
- Соответствие полю в другой таблице. Например, значение параметра "охват конференции" берется не из таблицы докладов на конференциях, а из таблицы конференций. Такое же соответствие для свойств журналов, которые связаны непосредственно с журналом, а не со статьями в них.
- Число записей, удовлетворяющих некоторому условию. Например, именно так считается число соавторов работы (это число записей авторства, связанных с данным объектом).

Генерация запросов к базе данных

При выполнении каждой строки формулы генерируется запрос на языке SQL. Каждый запрос возвращает восемь полей:

- ID автора (натуральное число);
- ID объекта работы (натуральное число);
- результат применения функции аппликации для объекта (действительное число);
- название работы (строка);
- некоторое свойство и его значение, например, "тип курса": "лекции" (строка + строка);
- число авторов работы (натуральное число);
- год, к которому относится работа.

На настоящее время генерация запросов основана на использовании шаблонов запросов и подстановке в них необходимых данных и ограничений. В частности, в каждом шаблоне определены название основной таблицы в базе данных, названия и условия для присоединяемых таблиц, если таковые имеются, соответствия параметров полям в базе данных и шаблоны для ограничений.

Модификаторы, как правило, не влияют на текст запроса, так как применяются к результату выполнения строки для объекта уже после того, как он посчитан. Однако иногда для упрощения имеет смысл включить модификатор непосредственно в текст запроса, как это сделано в примере ниже.

Приведем пример SQL-запроса для категории "авторство свидетельства о регистрации прав на программное обеспечение" (*patents.software*), без ограничений и с модификатором "разделить на число соавторов":

```
SELECT ap.f_man_id
     , ap.f_patprogram_id AS f_object_id
     , 1 / NVL(SUM(1) OVER (PARTITION BY ap.f_patprogram_id), 1) AS points
     , patprogram.f_patprogram_name AS act_name
     , 'Тип' AS act_paramname
     , 'Регистрация ПО' AS act_paramvalue
     , SUM(1) OVER (PARTITION BY ap.f_patprogram_id) AS act_nauthors
     , EXTRACT(year FROM patprogram.f_patprogram_date) AS year
FROM authorpog AS ap
JOIN patprogram ON (patprogram.f_patprogram_id = ap.f_patprogram_id)
WHERE EXTRACT(year FROM patprogram.f_patprogram_date) >= 2011
AND EXTRACT(year FROM patprogram.f_patprogram_date) <= 2015
```

Этот запрос может быть создан на основе, например, следующей JSON-строки: {"category": "patents.software", "modifiers": [["divide_coauthors"]]}.

В этом запросе основной таблицей является таблица AUTHORPOG, в которой хранятся авторства свидетельств о регистрации прав на программное обеспечение, и присоединяется таблица PATPROGRAM, в которой хранятся данные о самих свидетельствах. Присоединение нужно, чтобы проверить свойства свидетельства, в частности, год его публикации, и посчитать число совладельцев с помощью конструкции SUM(1) OVER PARTITION. Эта же конструкция используется при применении модификатора, на нее делится число баллов в поле *points*.

Рассмотрим более сложный пример запроса. Пусть JSON-строка выглядит так (категория "педагогическая работа"):

```
{"category": "study_work.type10", "property":
"total_hours_count", "educStage": 2}.
```

Запрос для этой строки имеет следующий вид:

```
SELECT ewm.f_man_id
     , ew.f_educwork_id AS f_object_id
     , NVL(f_educwork_week, 1) * NVL(f_educwork_hour, 1) points
     , NVL(ew.f_educwork_name, NVL(discipline.f_discipline_name, 'Без названия'))
         || ', ' || ew.f_educwork_year || ' уч. год'
         || ', ' || NVL(department.f_department_name, 'н/д')
         || (SELECT ', группы: ' || LISTAGG (f_educgroup_name, ', ') WITHIN GROUP
            FROM educgroup WHERE f_educwork_id = ew.f_educwork_id)
         || ' (ID: ' || few.f_feducwork_key || ')'
       AS act_name
     , 'Тип' AS act_paramname
     , et.f_eductype_name AS act_paramvalue
     , NVL(ewm.f_educworkman_part, 1) AS act_nauthors
     , ew.f_educwork_year AS year
FROM educwork ew
     JOIN eductype et ON (et.f_eductype_id = ew.f_eductype_id)
     JOIN educstage et ON (et.f_educstage_id = ew.f_educstage_id)
     JOIN educworkman ewm ON (ewm.f_educwork_id = ew.f_educwork_id)
     LEFT JOIN feducwork few ON (few.f_educwork_id = ew.f_educwork_id)
     LEFT JOIN department ON (ew.f_department_id = department.f_department_id)
     LEFT JOIN discipline ON (ew.f_discipline_id = discipline.f_discipline_id)
WHERE ew.f_educwork_year > = 2011 AND ew.f_educwork_year < = 2015
AND get_educwork_status(ew.f_educwork_id) > 0
AND et.f_eductype_id = 10
AND et.f_educstage_id = 2
```

Здесь присоединяется шесть таблиц, из которых три — в режиме "опционального" присоединения (конструкция LEFT JOIN). Эти таблицы в основном используются для составления поля с описанием (ACT_NAME). В шаблон здесь подставляется свойство "число часов в курсе", которое, как определено выше, является произведением значений двух полей. При наличии ограничений на любые параметры учебного курса они добавляются в конец шаблона. Здесь функция GET_EDUCWORK_STATUS проверяет, что запись проверена ответственным, а ограничения на F_EDUCTYPE_ID и F_EDUCSTAGE_ID проверяют соответственно, что тип курса — "семинары", а вид обучения — "бакалавр".

После выполнения запросов для каждой строки их результаты складываются во временную таблицу, затем суммирование баллов происходит из кода на языке Python.

Возможности улучшения генератора

Рассматривается возможность в будущем использовать отображение между реляционной базой данных и онтологиями для генерации запросов без использования шаблонов. Для этого планируется создать описания базы на языке R2RML [7], который является рекомендацией консорциума World Wide Web для задания соответствий между реляционной и онтологической базами данных.

Например, следующим образом может выглядеть фрагмент описания таблиц для свидетельств о регистрации прав на программное обеспечение:

```
<#SoftwarePatentsOwnership>
    rr:logicalTable [rr:tableName "authorpog"];
    rr:subjectMap [
        rr:template "http://istina.msu.ru/workers/{f_man_id}/";
        rr:class istina:worker;
    ];
```

```
    rr:predicateObjectMap [
       rr:predicate istina:authorOf;
       rr:objectMap [rr:template "http://istina.msu.ru/certificates/{f _ patprogram _ id}/"];
    ]
<#SoftwarePatents>
    rr:logicalTable [rr:tableName "patprogram"];
    rr:subjectMap [
       rr:template "http://istina.msu.ru/certificates/{f _ patprogram _ id}/";
       rr:class istina:patents.software;
    ];
    rr:predicateObjectMap [
       rr:predicate istina:registrationNumber;
       rr:objectMap [rr:column "f _ patprogram _ number"];
    ];
    rr:predicateObjectMap [
       rr:predicate istina:date;
       rr:objectMap [rr:column "f _ patprogram _ date"];
    ];
```

Здесь первая часть описывает отношение *authorOf* между авторами и свидетельствами, а вторая часть описывает сами свидетельства и их свойства. Например, на основании такого отображения могут быть сгенерированы следующие RDF-триплеты:

```
http://istina.msu.ru/workers/11634/authorOf http://istina.msu.ru/certificates/4602971/
http://istina.msu.ru/certificates/4602971/registrationNumber 2012618961
http://istina.msu.ru/certificates/4602971/date 2012-10-04
```

Заметим, что сгенерированные URL являются настоящими URL системы "ИСТИНА", доступными по HTTP. Это позволит также решать и обратную задачу: принимать от пользователя ссылку на объект, и анализировать, каким строкам формулы он мог бы подходить, и какие изменения необходимо внести, чтобы это было так.

В качестве альтернативы языку R2RML можно также использовать каноническое отображение реляционной базы в онтологии, описанное в работе [8].

Другой задачей, решению которой могла бы поспособствовать автоматическая генерация запросов, является поиск похожих строк и их объединение в одну строку. Например, если в формуле есть две строки, отличающиеся только диапазоном допустимых значений для некоторого параметра, то их можно объединить и заменить одним запросом, использующим конструкцию `CASE`.

Отдельной задачей является интеграция с онтологиями предметных областей. В случае наличия такой интеграции можно классифицировать работы по областям науки и учитывать в поиске только работы из области, описанной некоторым набором ключевых слов. С помощью такой интеграции также можно будет реализовать задачу информационного поиска [9].

Результаты апробации модели

Формулы расчета рейтинговых показателей, основанные на предложенной математической модели, используются в системе "ИСТИНА" с 2015 г. В рамках системы "ИСТИНА" был разработан графический интерфейс для редактирования формул, в котором доступны для использования все категории данных, имеющиеся в системе. Категории разбиты на несколько блоков, таких как "Научная работа", "Учебная работа", "Финансирование" и "Музейная работа".

Функционально подсистема разделена на несколько частей в зависимости от статуса пользователя.

Для лиц от структурных подразделений, ответственных за сопровождение данных в системе "ИСТИНА", доступен интерфейс, позволяющий создавать формулы, редактировать их, привязывать их к подразделениям (факультетам, кафедрам, институтам) и должностям (научные сотрудники, профессорско-преподавательский состав), а также публиковать (делать общедоступными). Ответственные лица могут также смотреть результаты выполнения формулы для своего подразделения и подробные отчеты для каждого сотрудника подразделения.

Каждый пользователь системы может просмотреть любую опубликованную формулу и посмотреть результат ее выполнения для своих работ с возможностью экспорта в формат PDF.

Большинство страниц подсистемы генерируются на стороне сервера с помощью web-фреймворка Django. Генерацию содержимого на стороне клиента использует только редактор формул, большая часть которого написана на языке JavaScript. Подробно архитектура системы изложена в работе [10].

Система показала свою актуальность и востребованность в МГУ имени М. В. Ломоносова, а также в ряде других организаций, которые приступили к ее использованию в своих интересах. Разрабатывается интеграция подсистемы формул с другими разделами системы, такими как учебные планы и конкурсы.

Список литературы

1. **Система** Elsevier Pure. URL: https://www.elsevier.com/solutions/pure
2. **Система** Thomson Reuters Converis. URL: http://converis.thomsonreuters.com/
3. **Электронный** научный архив Уральского федерального университета. URL: http://elar.urfu.ru/
4. **Информационно**-аналитическая система Астраханского государственного университета. URL: http://science.asu.edu.ru/
5. **Информационно**-аналитическая система сопровождения научно-исследовательской деятельности Санкт-Петербургского государственного университета. URL: https://ias.spbu.ru/
6. **Загорулько Ю. А., Загорулько Г. Б., Боровикова О. И.** Технология создания тематических интеллектуальных научных интернет-ресурсов, базирующаяся на онтологии // Программная инженерия. 2016. № 2. С. 51—60.
7. **R2RML:** RDB to RDF Mapping Language (рекомендация W3C от 27.09.2012). URL: https://www.w3.org/TR/r2rml/
8. **A Direct** Mapping of Relational Data to RDF (рекомендация W3C от 27.09.2012). URL: https://www.w3.org/TR/r2rml/
9. **Афонин С. А., Козицын А. С.** Использование онтологий в поисковых системах // Материалы Всероссийской конференции с международным участием "Знания-Онтологии-Теории", г. Новосибирск, 2009, Т. 2. С. 47—52.
10. **Садовничий В. А., Афонин С. А., Бахтин А. В.** и др. Интеллектуальная система тематического исследования научно-технической информации (ИСТИНА). М.: Изд-во Мос. гос. ун-та, 2014. 262 с.

Software Mechanisms for Scientometrical Data Aggregation Based on Ontological Representation of the Relational Database Structure

S. A. Afonin, serg@msu.ru, **A. S. Kozitsyn**, alexanderkz@mail.ru, **D. A. Shachnev**, mitya57@mitya57.me, Moscow State University, Moscow, 119991, Russian Federation

Corresponding author:

Shachnev Dmitry A., Postgraduate Student, Moscow State University, Moscow, 119991, Russian Federation, e-mail: mitya57@mitya57.me

Received on May 19, 2016
Accepted on May 30, 2016

The article presents a mathematical model and a software implementation of the aggregator of scientometrical data which describes the results of individuals' activities. The model is based on using the ontological structure of data categories and on summation of values of different parameters (indicators) which define various components of the activity measuring, for each of these categories. Based on calculation formulas the SQL queries to the relational database are built, which are used to fill the table with the activity results corresponding to the formula. The model can be used for finding the individuals or groups that have shown the best results in a certain area, and for collecting the statistics about the departments' activities. As an example of database structure is used the structure of the ISTINA informational-analytical system (Intellectual System of Thematic Examination of Scientometrical data) which has been developed and successfully exploited for three years at Lomonosov Moscow State University. The data aggregation mechanisms have been successfully tested on that system.

Keywords: *ontologies, Semantic Web, scientometrical data, SQL query generation, data aggregation*

For citation:

Afonin S. A., Kozitsyn A. S., Shachnev D. A. Software Mechanisms for Scientometrical Data Aggregation Based on Ontological Representation of the Relational Database Structure, *Programmnaya Ingeneria*, 2016, vol. 7, no. 9, pp. 408—413.

DOI: 10.17587/prin.7.408-413

References

1. **Elsevier** Pure, available at: https://www.elsevier.com/solutions/pure
2. **Thomson** Reuters Converis, available at: http://converis.thomsonreuters.com/
3. **Electronic** Science Archive of Ural Federal University, available at: http://elar.urfu.ru/
4. **Information** analytics system of Astrahkan State University, available at: http://science.asu.edu.ru/
5. **Information** analytics system for supporting scientific and research work of Saint Petersburg State University, available at: https://ias.spbu.ru/
6. **Zagorulko Yu. A., Zagorulko G. B., Borovikova O. I.** Tehnologija sozdanija tematicheskih intellektual'nyh nauchnyh internet-resursov, bazirujushhajasja na ontologii **(**Technology for creating thematic intellectual scientific internet resources based on ontology), *Programmnaya Ingeneria*, 2016, vol. 7, no. 2, pp. 51—60 (in Russian).
7. **R2RML**: RDB to RDF Mapping Language (W3C recommendation from 27.09.2012), available at: https://www.w3.org/TR/r2rml/
8. **A Direct** Mapping of Relational Data to RDF (W3C recommendation from 27.09.2012), available at: https://www.w3.org/TR/r2rml/
9. **Afonin S. A., Kozitsyn A. S.** Ispol'zovanie ontologij v poiskovyh sistemah (Use of ontologies in search engine)s, *Knowledge-Ontologies-Theories" conference proceedings*, Novosibirsk, 2009, vol. 2, pp. 47—52 (in Russian).
10. **Sadovnichiy V. A., Afonin S.A, Bakhtin A. V.** et al. *Intellektual'naja sistema tematicheskogo issledovanija nauchno-tehnicheskoj informacii (ISTINA)* (Intellectual System of Thematic Examination of Scientific and Technical Data (ISTINA)), Moscow, Moscow State University Publishing, 2014, 262 p. (in Russian).

УДК 629.73.02; 629.73.05/.06; 535.643 DOI: 10.17587/prin.7.414-423

И. О. Жаринов, д-р техн. наук, доц., зав. каф., Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики (Университет ИТМО), АО "ОКБ "Электроавтоматика", e-mail: igor_rabota@pisem.net,
О. О. Жаринов, канд. техн. наук, доц., Санкт-Петербургский государственный университет аэрокосмического приборостроения (ГУАП), e-mail: zharinov73@hotbox.ru

Способы оценки коэффициентов матрицы профиля экранов с трехкомпонентной схемой цветовоспроизведения

Рассмотрена задача оценки коэффициентов матрицы профиля экранов с трехкомпонентной схемой цветовоспроизведения. Трехкомпонентная схема цветовоспроизведения предполагает формирование цвета пикселя экрана с использованием трех источников света. Источниками света являются светодиоды красного, зеленого и синего цветов или жидкие кристаллы, поляризующие белый свет от системы подсвета экрана. Матрица профиля экрана связывает значения кодов RGB и координаты цвета XYZ выводимого на экран пикселя изображения и определяет цветовой охват экрана. Изготовители экранов не приводят в технической документации числовые значения коэффициентов матрицы профиля, между тем эти данные необходимы для управления цветом элементов изображения. Предложено семь различных способов оценки коэффициентов матрицы профиля экрана, которые можно разделить на две группы. Исходными данными для первой группы способов являются координаты цветности вершин треугольника цветового охвата и координаты цветности точки белого цвета экрана, заданные в технической документации на экран. Исходными данными для второй группы способов являются результаты колориметрических экспериментов, представленные в параметрах различных цветовых моделей. Инструментальные средства, необходимые для оценки коэффициентов матрицы профиля экранов, должны обеспечивать измерения в параметрах хотя бы в одной из стандартизованных цветовых моделей: XYZ, Yxy, Yu′v′.

Ключевые слова: *цветовые пространства, матрица профиля, экран, колориметрические измерения, координаты цвета, координаты цветности*

Введение

В процессе проектирования компонентов человеко-машинных интерфейсов разработчики сталкиваются с необходимостью математического описания колориметрических свойств экранов. Колориметрические свойства экранов определяют особенности их цветовоспроизведения.

Цветовоспроизведение экранов основано на аддитивном законе смешения цветов Грассмана. Цвет пикселя в современных экранах формируется по трехкомпонентному принципу — совместным свечением трех составляющих основных цветов: красного, зеленого и синего. Источниками свечения могут быть элементарные цветные светодиоды или жидкие кристаллы, поляризующие белый свет от системы подсвета экрана.

На программном уровне цвет свечения каждого пикселя экрана задается кодом RGB (R — Red, G — Green, B — Blue), составляющие которого определяют интенсивность свечения соответственно красного, зеленого и синего цветов.

Математическое описание цветовоспроизведения экранов определяется коэффициентами X_r, Y_r, Z_r, X_g, Y_g, Z_g, X_b, Y_b, Z_b матрицы профиля экрана. Матрица профиля экрана связывает значения кодов RGB и координаты цвета XYZ выводимого на экран изображения [1]:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}, \quad \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{bmatrix}^{-1} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}. \quad (1)$$

Координаты цвета XYZ являются исходными данными для большинства методик выполнения колориметрических расчетов, а значения коэффициентов матрицы профиля полностью определяют цветовой охват экрана, т. е. тот диапазон цветов и цветовых оттенков, которые могут быть воспроизведены на этом экране.

Цветовой охват экрана с трехкомпонентной схемой формирования цвета аппроксимируется на цветовой плоскости xy или $u'v'$ треугольником. Переход от ко-

ординат цвета XYZ к (x,y)- или (u', v')-координатам цветности осуществляется по формулам [2], введенным Международной комиссией по освещению:

$$x = \frac{X}{X+Y+Z},\ y = \frac{Y}{X+Y+Z},\ X = \frac{xY}{y},$$
$$Z = (1-x-y)\frac{Y}{y},\ u' = \frac{4X}{X+15Y+3Z},$$
$$v' = \frac{9Y}{X+15Y+3Z},\ X = \frac{9Yu'}{4v'},\ Z = \frac{3Y(4-u')}{4v'} - 5Y, \quad (2)$$
$$u' = \frac{2x}{6y-x+1,5},\ v' = \frac{4,5y}{6y-x+1,5},$$
$$x = \frac{4,5u'}{3u'-8v'+6},\ y = \frac{2v'}{3u'-8v'+6}.$$

Разработчики экранов не приводят в технической документации числовые значения коэффициентов X_r, Y_r, Z_r, X_g, Y_g, Z_g, X_b, Y_b, Z_b матрицы профиля. В ряде случаев документация содержит лишь числовые значения (x, y)- или (u', v')-координат цветности вершин треугольника цветового охвата экрана.

В связи с этим актуальной является задача вывода математических выражений, позволяющих получать оценки $\widehat{X}_r$, $\widehat{Y}_r$, $\widehat{Z}_r$, $\widehat{X}_g$, $\widehat{Y}_g$, $\widehat{Z}_g$, $\widehat{X}_b$, $\widehat{Y}_b$, $\widehat{Z}_b$ значений коэффициентов матрицы профиля экранов.

1. Преобразование цветовых моделей

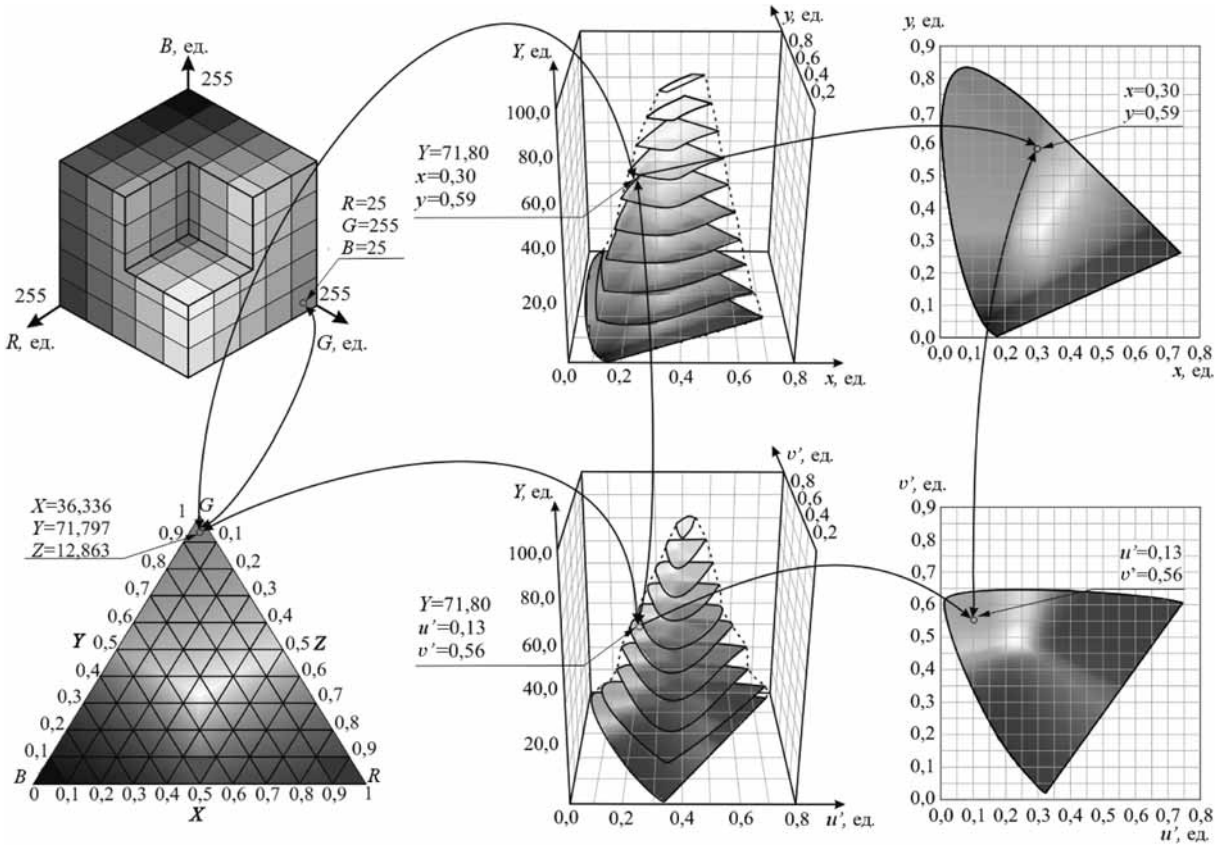
В основе математических выражений для вычисления оценок $\widehat{X}_r$, $\widehat{Y}_r$, $\widehat{Z}_r$, $\widehat{X}_g$, $\widehat{Y}_g$, $\widehat{Z}_g$, $\widehat{X}_b$, $\widehat{Y}_b$, $\widehat{Z}_b$ коэффициентов матрицы профиля экранов лежат базовые преобразования (1), (2), связывающие цветовые координаты в различных цветовых моделях.

Общие правила преобразования цветовых моделей, актуальные для решения задачи оценки коэффициентов матрицы профиля экранов, показаны на рисунке. В каждой цветовой модели любой цвет представляется точкой, принадлежащей цветовому телу в многомерном цветовом пространстве, оси которого определены цветовыми координатами.

Существуют следующие способы вычисления оценок $\widehat{X}_r$, $\widehat{Y}_r$, $\widehat{Z}_r$, $\widehat{X}_g$, $\widehat{Y}_g$, $\widehat{Z}_g$, $\widehat{X}_b$, $\widehat{Y}_b$, $\widehat{Z}_b$ коэффициентов матрицы профиля экранов:

- на основе колориметрических расчетов и анализа данных, представленных разработчиком в технической документации на экран;
- на основе колориметрических экспериментов с использованием контрольно-измерительной аппаратуры, позволяющей получать значения цветовых координат в различных цветовых моделях.

Колориметрические расчеты и анализ данных, представленных в технической документации на экран, предполагают использование в качестве ис-



Правила преобразования цветовых моделей

ходных данных значения координат цветности вершин треугольника цветового охвата экрана и координат цветности точки белого цвета экрана, заданных в параметрах различных цветовых моделей.

В колориметрических экспериментах традиционно используют [3—8]:

- колориметр — оптический прибор для измерения интенсивности цвета в какой-либо цветовой модели;
- спектрорадиометр — оптический прибор, предназначенный для измерения фотометрических характеристик (световой поток, освещенность, сила света, яркость и т. п.) источников оптического излучения;
- спектрометр — оптический прибор, используемый в спектроскопических исследованиях для накопления спектра, его количественной обработки и последующего анализа с помощью математических методов;
- спектрофотометр — оптический прибор, предназначенный для измерения отношений двух потоков оптического излучения, один из которых — поток, падающий на исследуемый образец, другой — поток, испытавший взаимодействие с образцом; в результате измерений получается спектр отношений потоков.

Могут использоваться и другие колориметрические приборы. Известны, в частности, разработки оптических измерительных приборов фирм Carry, N-Vision, Minolta, ООО "НПП "ТКА", АНО НТЦСЭ "ИСЭП", ФГУП "ВНИИОФИ".

2. Оценка коэффициентов профиля экрана по данным технической документации, заданным на плоскости *xy*

Исходными данными для оценки коэффициентов профиля экрана являются представленные разработчиком в технической документации значения координат цветности (x_R, y_R), (x_G, y_G), (x_B, y_B) вершин треугольника цветового охвата в красном *R*, зеленом *G* и синем *B* цвете соответственно, а также значения координат цветности точки белого цвета (x_W, y_W), соответствующей экрану [9, 10].

В этом случае оценка коэффициентов $\hat{X}_r$, $\hat{Y}_r$, $\hat{Z}_r$, $\hat{X}_g$, $\hat{Y}_g$, $\hat{Z}_g$, $\hat{X}_b$, $\hat{Y}_b$, $\hat{Z}_b$ профиля экрана осуществляется путем решения системы уравнений

$$\begin{cases} x_R\left(\hat{X}_r+\hat{Y}_r+\hat{Z}_r\right)=\hat{X}_r=\dfrac{x_R}{y_R}\hat{Y}_r,\ y_R\left(\hat{X}_r+\hat{Y}_r+\hat{Z}_r\right)=\hat{Y}_r=\dfrac{y_R}{x_R}\hat{X}_r,\ x_G\left(\hat{X}_g+\hat{Y}_g+\hat{Z}_g\right)=\hat{X}_g=\dfrac{x_G}{y_G}\hat{Y}_g, \\ y_G\left(\hat{X}_g+\hat{Y}_g+\hat{Z}_g\right)=\hat{Y}_g=\dfrac{y_G}{x_G}\hat{X}_g,\ x_B\left(\hat{X}_b+\hat{Y}_b+\hat{Z}_b\right)=\hat{X}_b=\dfrac{x_B}{y_B}\hat{Y}_b,\ y_B\left(\hat{X}_b+\hat{Y}_b+\hat{Z}_b\right)=\hat{Y}_b\dfrac{y_B}{x_B}\hat{X}_b, \\ x_W\left(\hat{X}_r+\hat{X}_g+\hat{X}_b+\hat{Y}_r+\hat{Y}_g+\hat{Y}_b+\hat{Z}_r+\hat{Z}_g+\hat{Z}_b\right)=\hat{X}_r+\hat{X}_g+\hat{X}_b, \\ y_W\left(\hat{X}_r+\hat{X}_g+\hat{X}_b+\hat{Y}_r+\hat{Y}_g+\hat{Y}_b+\hat{Z}_r+\hat{Z}_g+\hat{Z}_b\right)=\hat{Y}_r+\hat{Y}_g+\hat{Y}_b. \end{cases} \tag{3}$$

Уравнения системы (3) получены путем преобразования уравнений (1), (2). Важно заметить, что координаты цветности (x_R, y_R), (x_G, y_G), (x_B, y_B) соответствуют максимально насыщенным основным цветам, заданным кодами *RGB*, т. е. (x_R, y_R) соответствует код (255, 0, 0), (x_G, y_G) соответствует код (0, 255, 0), (x_B, y_B) соответствует код (0, 0, 255). Координаты цветности (x_W, y_W) точки белого цвета соответствуют коду (255, 255, 255). Значение 255 является максимальным для кода одного основного цвета 8-битной модели *RGB*.

Система (3) включает восемь уравнений и имеет девять неизвестных: $\hat{X}_r$, $\hat{Y}_r$, $\hat{X}_g$, $\hat{Y}_g$, $\hat{X}_b$, $\hat{Y}_b$, $\hat{Z}_r$, $\hat{Z}_g$, $\hat{Z}_b$. Для решения системы (3) используется девятое уравнение

$$\hat{Y}_r+\hat{Y}_g+\hat{Y}_b=1, \tag{4}$$

описывающее баланс белого цвета экрана.

Совместное решение уравнений (3), (4) имеет следующий вид:

$$\begin{gathered} \hat{X}_r=\frac{x_R}{y_R}\hat{Y}_r,\ \hat{X}_g=\frac{x_G}{y_G}\hat{Y}_g,\ \hat{X}_b=\frac{x_B}{y_B}\hat{Y}_b,\ \hat{Y}_r=1-\hat{Y}_g-\hat{Y}_b,\ \hat{Y}_g=\frac{\dfrac{x_W}{y_W}-\dfrac{x_R}{y_R}+\left(\dfrac{x_R}{y_R}-\dfrac{x_B}{y_B}\right)\hat{Y}_b}{\dfrac{x_g}{y_g}-\dfrac{x_R}{y_R}}, \\ \hat{Y}_b=\frac{\dfrac{1-x_W}{y_W}+\dfrac{x_R-1}{y_R}+\dfrac{x_Wy_R-y_Wx_R}{x_Gy_R-y_Gx_R}\cdot\dfrac{y_G}{y_W}\cdot\left(\dfrac{1-x_R}{y_R}+\dfrac{x_G-1}{y_G}\right)}{\dfrac{x_Ry_B-y_Rx_B}{x_Gy_R-y_Gx_R}\cdot\dfrac{y_G}{y_B}\cdot\left(\dfrac{1-x_G}{y_G}+\dfrac{x_R-1}{y_R}\right)+\dfrac{1-x_B}{y_B}+\dfrac{x_R-1}{y_R}}, \\ \hat{Z}_r=\hat{Y}_r\left(\frac{1-x_R-y_R}{y_R}\right),\ \hat{Z}_g=\hat{Y}_g\left(\frac{1-x_G-y_G}{y_G}\right),\ \hat{Z}_b=\hat{Y}_b\left(\frac{1-x_B-y_B}{y_B}\right) \end{gathered} \tag{5}$$

Выражения (5) позволяют вычислять оценки коэффициентов $\hat{X}_r$, $\hat{Y}_r$, $\hat{Z}_r$, $\hat{X}_g$, $\hat{Y}_g$, $\hat{Z}_g$, $\hat{X}_b$, $\hat{Y}_b$, $\hat{Z}_b$ профиля экрана по данным технической документации, заданным на плоскости *xy*. Первым вычисляют коэффициент $\hat{Y}_b$, далее вычисляют коэффициенты $\hat{Y}_g$, $\hat{Y}_r$, на основе которых вычисляют коэффициенты $\hat{X}_r$, $\hat{Z}_r$, $\hat{X}_g$, $\hat{Z}_g$, $\hat{X}_b$, $\hat{Z}_b$.

3. Оценка коэффициентов профиля экрана по данным технической документации, заданным на плоскости *u′v′*

Исходными данными для оценки коэффициентов профиля экрана являются представленные разработчиком в технической документации значения координат цветности (u'_R, v'_R), (u'_G, v'_G), (u'_B, v'_B) вершин треугольника цветового охвата в красном (*R*), зеленом (*G*) и синем (*B*) цвете соответственно, а также значения координат цветности точки белого цвета (u'_W, v'_W), соответствующей экрану [11].

В этом случае оценки коэффициентов $\widehat{X}_r$, $\widehat{Y}_r$, $\widehat{Z}_r$, $\widehat{X}_g$, $\widehat{Y}_g$, $\widehat{Z}_g$, $\widehat{X}_b$, $\widehat{Y}_b$, $\widehat{Z}_b$ профиля экрана имеют вид:

$$\widehat{X}_r = \frac{9u'_R}{4v'_R}\widehat{Y}_r,\ \widehat{X}_g = \frac{9u'_G}{4v'_G}\widehat{Y}_G,\ \widehat{X}_b = \frac{9u'_B}{4v'_B}\widehat{Y}_B,\ \widehat{Y}_r = 1 - \widehat{Y}_g - \widehat{Y}_b,\ \widehat{Z}_g = \frac{3\widehat{Y}_g(4-u'_G)}{4v'_G} - 5\widehat{Y}_g,$$

$$\widehat{Y}_g = \frac{\det\begin{pmatrix} u'_W & u'_R \\ v'_W & v'_R \end{pmatrix} - \widehat{Y}_b \det\begin{pmatrix} u'_B & u'_R \\ v'_B & v'_R \end{pmatrix}}{\det\begin{pmatrix} u'_G & u'_R \\ v'_G & v'_R \end{pmatrix}},\ \widehat{Z}_r = \frac{3\widehat{Y}_r(4-u'_R)}{4v'_R} - 5\widehat{Y}_r,\ \widehat{Z}_b = \frac{3\widehat{Y}_b(4-u'_B)}{4v'_B} - 5\widehat{Y}_b,$$

$$\widehat{Y}_b = \frac{v'_B}{v'_W}\frac{\det\begin{pmatrix} \det\begin{pmatrix} u'_G & u'_R \\ v'_G & v'_R \end{pmatrix} & \det\begin{pmatrix} 4-u'_G & v'_G \\ 4-u'_R & v'_R \end{pmatrix} \\ \det\begin{pmatrix} u'_W & u'_R \\ v'_W & v'_R \end{pmatrix} & \det\begin{pmatrix} 4-u'_W & v'_W \\ 4-u'_R & v'_R \end{pmatrix} \end{pmatrix}}{\det\begin{pmatrix} \det\begin{pmatrix} u'_G & u'_R \\ v'_G & v'_R \end{pmatrix} & \det\begin{pmatrix} 4-u'_G & v'_G \\ 4-u'_R & v'_R \end{pmatrix} \\ \det\begin{pmatrix} u'_B & u'_R \\ v'_B & v'_R \end{pmatrix} & \det\begin{pmatrix} 4-u'_B & v'_B \\ 4-u'_R & v'_R \end{pmatrix} \end{pmatrix}}. \quad (6)$$

Выражения (6) получены в результате решения системы уравнений:

$$\begin{cases} u'_R\left(\widehat{X}_r + \widehat{Y}_r + \widehat{Z}_r\right) = 4\widehat{X}_r = 4\dfrac{9\widehat{Y}_r u'_R}{v'_R},\ v'_R\left(\widehat{X}_r + \widehat{Y}_r + \widehat{Z}_r\right) = 9\widehat{Y}_r = \dfrac{\widehat{X}_r v'_R}{u'_R},\ u'_G\left(\widehat{X}_g + \widehat{Y}_g + \widehat{Z}_g\right) = 4\widehat{X}_g = 4\dfrac{9\widehat{Y}_g u'_G}{v'_G}, \\ v'_G\left(\widehat{X}_g + \widehat{Y}_g + \widehat{Z}_g\right) = 9\widehat{Y}_g = \dfrac{\widehat{X}_g v'_G}{u'_G},\ u'_B\left(\widehat{X}_b + \widehat{Y}_b + \widehat{Z}_b\right) = 4\widehat{X}_b = 4\dfrac{9\widehat{Y}_b u'_B}{v'_B},\ v'_B\left(\widehat{X}_b + \widehat{Y}_b + \widehat{Z}_b\right) = 9\widehat{Y}_b = \dfrac{\widehat{X}_b v'_B}{u'_B}, \\ u'_W = \dfrac{4\left(\widehat{X}_r + \widehat{X}_g + \widehat{X}_b\right)}{\left(\widehat{X}_r + \widehat{X}_g + \widehat{X}_b + 15\left(\widehat{Y}_r + \widehat{Y}_g + \widehat{Y}_b\right) + 3\left(\widehat{Z}_r + \widehat{Z}_g + \widehat{Z}_b\right)\right)}, \\ v'_W = \dfrac{9\left(\widehat{Y}_r + \widehat{Y}_g + \widehat{Y}_b\right)}{\left(\widehat{X}_r + \widehat{X}_g + \widehat{X}_b + 15\left(\widehat{Y}_r + \widehat{Y}_g + \widehat{Y}_b\right) + 3\left(\widehat{Z}_r + \widehat{Z}_g + \widehat{Z}_b\right)\right)}, \\ \widehat{Y}_r + \widehat{Y}_g + \widehat{Y}_b = 1. \end{cases} \quad (7)$$

Координаты цветности (u'_R, v'_R), (u'_G, v'_G), (u'_B, v'_B) также соответствуют максимально насыщенным основным цветам, заданным кодами *RGB*, т. е. (u'_R, v'_R) соответствует код (255, 0, 0), (u'_G, v'_G) соответствует код (0, 255, 0), (u'_B, v'_B) соответствует код (0, 0, 255). Координаты цветности (u'_W, v'_W) точки белого цвета соответствуют коду (255, 255, 255).

Важно заметить, что оценки коэффициентов $\widehat{X}_r$, $\widehat{Y}_r$, $\widehat{Z}_r$, $\widehat{X}_g$, $\widehat{Y}_g$, $\widehat{Z}_g$, $\widehat{X}_b$, $\widehat{Y}_b$, $\widehat{Z}_b$, полученные с помощью выражений (5) и (6), будут соответствовать типовым значениям координат цветности (x_R, y_R), (x_G, y_G), (x_B, y_B), (x_W, y_W), (u'_R, v'_R), (u'_G, v'_G), (u'_B, v'_B), (u'_W, v'_W) вершин треугольника цветового охвата экрана. Вместе с этим, для описания характеристик технологического разброса координат цветности экранов разработчики экранов приводят в технической документации также минимальные и максимальные значения для каждой из этих величин.

4. Оценка коэффициентов профиля экрана по результатам измерений, полученным в параметрах цветовой плоскости *xy*

В случае, когда координаты цветности (x_R, y_R), (x_G, y_G), (x_B, y_B) вершин треугольника цветового охвата и координаты цветности (x_W, y_W) точки белого цвета не представлены разработчиком в технической документации экрана, оценки коэффициентов профиля экрана могут быть получены [9, 10] в результате экспериментальных измерений колориметрических характеристик экрана в параметрах цветовой плоскости *xy*.

Для этого необходимо отобразить на экране произвольным образом заданные кодами (R_1, G_1, B_1), (R_2, G_2, B_2), (R_3, G_3, B_3), (R_4, G_4, B_4) четыре цвета и провести измерения соответствующих им координат цветности (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4).

В этом случае оценки коэффициентов $\widehat{X}_r$, $\widehat{Y}_r$, $\widehat{Z}_r$, $\widehat{X}_g$, $\widehat{Y}_g$, $\widehat{Z}_g$, $\widehat{X}_b$, $\widehat{Y}_b$, $\widehat{Z}_b$ профиля экрана будут иметь следующий вид:

$$\begin{cases} \begin{bmatrix} \widehat{X}_r \\ \widehat{X}_g \\ \widehat{X}_b \\ \widehat{Y}_g \\ \widehat{Y}_b \\ \widehat{Z}_r \\ \widehat{Z}_g \\ \widehat{Z}_b \end{bmatrix} = \begin{bmatrix} d_{11} & d_{12} & d_{13} & d_{14} & d_{15} & d_{16} & d_{17} & d_{18} \\ d_{21} & d_{22} & d_{23} & d_{24} & d_{25} & d_{26} & d_{27} & d_{28} \\ d_{31} & d_{32} & d_{33} & d_{34} & d_{35} & d_{36} & d_{37} & d_{38} \\ d_{41} & d_{42} & d_{43} & d_{44} & d_{45} & d_{46} & d_{47} & d_{48} \\ d_{51} & d_{52} & d_{53} & d_{54} & d_{55} & d_{56} & d_{57} & d_{58} \\ d_{61} & d_{62} & d_{63} & d_{64} & d_{65} & d_{66} & d_{67} & d_{68} \\ d_{71} & d_{72} & d_{73} & d_{74} & d_{75} & d_{76} & d_{77} & d_{78} \\ d_{81} & d_{82} & d_{83} & d_{84} & d_{85} & d_{86} & d_{87} & d_{88} \end{bmatrix}^{-1} \begin{bmatrix} x_1 R_1 \\ R_1(1-y_1) \\ x_2 R_2 \\ R_2(1-y_2) \\ x_3 R_3 \\ R_3(1-y_3) \\ x_4 R_4 \\ R_4(1-y_4) \end{bmatrix}, \\ Y_r = 1 - \widehat{Y}_g - \widehat{Y}_b \end{cases} \tag{8}$$

где $d_{11} = R_1(1-x_1)$, $d_{12} = G_1(1-x_1)$, $d_{13} = B_1(1-x_1)$, $d_{14} = x_1(R_1 - G_1)$, $d_{15} = x_1(R_1 - B_1)$, $d_{16} = -x_1R_1$, $d_{17} = -x_1G_1$, $d_{18} = -x_1B_1$, $d_{21} = R_1y_1$, $d_{22} = G_1y_1$, $d_{23} = B_1y_1$, $d_{24} = (R_1 - G_1)(1-y_1)$, $d_{25} = (R_1 - B_1)(1-y_1)$, $d_{26} = R_1y_1$, $d_{27} = G_1y_1$, $d_{28} = B_1y_1$, $d_{31} = R_2(1-x_2)$, $d_{32} = G_2(1-x_2)$, $d_{33} = B_2(1-x_2)$, $d_{34} = x_2(R_2 - G_2)$, $d_{35} = x_2(R_2 - B_2)$, $d_{36} = -x_2R_2$, $d_{37} = -x_2G_2$, $d_{38} = -x_2B_2$, $d_{41} = R_2y_2$, $d_{42} = G_2y_2$, $d_{43} = B_2y_2$, $d_{44} = (R_2 - G_2)(1-y_2)$, $d_{45} = (R_2 - B_2)(1-y_2)$, $d_{46} = R_2y_2$, $d_{47} = G_2y_2$, $d_{48} = B_2y_2$, $d_{51} = R_3(1-x_3)$, $d_{52} = G_3(1-x_3)$, $d_{53} = B_3(1-x_3)$, $d_{54} = x_3(R_3 - G_3)$, $d_{55} = x_3(R_3 - B_3)$, $d_{56} = -x_3R_3$, $d_{57} = -x_3G_3$, $d_{58} = -x_3B_3$, $d_{61} = R_3y_3$, $d_{62} = G_3y_3$, $d_{63} = B_3y_3$, $d_{64} = (R_3 - G_3)(1-y_3)$, $d_{65} = (R_3 - B_3)(1-y_3)$, $d_{66} = R_3y_3$, $d_{67} = G_3y_3$, $d_{68} = B_3y_3$, $d_{71} = R_4(1-x_4)$, $d_{72} = G_4(1-x_4)$, $d_{73} = B_4(1-x_4)$, $d_{74} = x_4(R_4 - G_4)$, $d_{75} = x_4(R_4 - B_4)$, $d_{76} = -x_4R_4$, $d_{77} = -x_4G_4$, $d_{78} = -x_4B_4$, $d_{81} = R_4y_4$, $d_{82} = G_4y_4$, $d_{83} = B_4y_4$, $d_{84} = (R_4 - G_4)(1-y_4)$, $d_{85} = (R_4 - B_4)(1-y_4)$, $d_{86} = R_4y_4$, $d_{87} = G_4y_4$, $d_{88} = B_4y_4$.

Система (8) получена путем решения системы уравнений

$$\begin{cases} \widehat{X}_rR_1 + \widehat{X}_gG_1 + \widehat{X}_bB_1 = x_1\left(\left(\widehat{X}_r + \widehat{Y}_r + \widehat{Z}_r\right)R_1 + \left(\widehat{X}_g + \widehat{Y}_g + \widehat{Z}_g\right)G_1 + \left(\widehat{X}_b + \widehat{Y}_b + \widehat{Z}_b\right)B_1\right), \\ \widehat{Y}_rR_1 + \widehat{Y}_gG_1 + \widehat{Y}_bB_1 = y_1\left(\left(\widehat{X}_r + \widehat{Y}_r + \widehat{Z}_r\right)R_1 + \left(\widehat{X}_g + \widehat{Y}_g + \widehat{Z}_g\right)G_1 + \left(\widehat{X}_b + \widehat{Y}_b + \widehat{Z}_b\right)B_1\right), \\ \widehat{X}_rR_2 + \widehat{X}_gG_2 + \widehat{X}_bB_2 = x_2\left(\left(\widehat{X}_r + \widehat{Y}_r + \widehat{Z}_r\right)R_2 + \left(\widehat{X}_g + \widehat{Y}_g + \widehat{Z}_g\right)G_2 + \left(\widehat{X}_b + \widehat{Y}_b + \widehat{Z}_b\right)B_2\right), \\ \widehat{Y}_rR_2 + \widehat{Y}_gG_2 + \widehat{Y}_bB_2 = y_2\left(\left(\widehat{X}_r + \widehat{Y}_r + \widehat{Z}_r\right)R_2 + \left(\widehat{X}_g + \widehat{Y}_g + \widehat{Z}_g\right)G_2 + \left(\widehat{X}_b + \widehat{Y}_b + \widehat{Z}_b\right)B_2\right), \\ \widehat{X}_rR_3 + \widehat{X}_gG_3 + \widehat{X}_bB_3 = x_3\left(\left(\widehat{X}_r + \widehat{Y}_r + \widehat{Z}_r\right)R_3 + \left(\widehat{X}_g + \widehat{Y}_g + \widehat{Z}_g\right)G_3 + \left(\widehat{X}_b + \widehat{Y}_b + \widehat{Z}_b\right)B_3\right), \\ \widehat{Y}_rR_3 + \widehat{Y}_gG_3 + \widehat{Y}_bB_3 = y_3\left(\left(\widehat{X}_r + \widehat{Y}_r + \widehat{Z}_r\right)R_3 + \left(\widehat{X}_g + \widehat{Y}_g + \widehat{Z}_g\right)G_3 + \left(\widehat{X}_b + \widehat{Y}_b + \widehat{Z}_b\right)B_3\right), \\ \widehat{X}_rR_4 + \widehat{X}_gG_4 + \widehat{X}_bB_4 = x_4\left(\left(\widehat{X}_r + \widehat{Y}_r + \widehat{Z}_r\right)R_4 + \left(\widehat{X}_g + \widehat{Y}_g + \widehat{Z}_g\right)G_4 + \left(\widehat{X}_b + \widehat{Y}_b + \widehat{Z}_b\right)B_4\right), \\ \widehat{Y}_rR_4 + \widehat{Y}_gG_4 + \widehat{Y}_bB_4 = y_4\left(\left(\widehat{X}_r + \widehat{Y}_r + \widehat{Z}_r\right)R_4 + \left(\widehat{X}_g + \widehat{Y}_g + \widehat{Z}_g\right)G_4 + \left(\widehat{X}_b + \widehat{Y}_b + \widehat{Z}_b\right)B_4\right), \\ \widehat{Y}_r = 1 - \widehat{Y}_g - \widehat{Y}_b. \end{cases}$$

5. Оценка коэффициентов профиля экрана по результатам измерений, полученным в параметрах цветовой плоскости *u′v′*

В случае, когда координаты цветности (u'_R, v'_R), (u'_G, v'_G), (u'_B, v'_B) вершин треугольника цветового охвата и координаты цветности (u'_W, v'_W) точки белого цвета не представлены разработчиком в технической документации экрана, оценки коэффициентов профиля экрана могут быть получены [11] в результате экспериментальных измерений колориметрических характеристик экрана в параметрах цветовой плоскости $u'v'$.

Для этого необходимо отобразить на экране произвольным образом заданные кодами (R_1, G_1, B_1), (R_2, G_2, B_2), (R_3, G_3, B_3), (R_4, G_4, B_4) четыре цвета и провести измерения соответствующих им координат цветности (u'_R, v'_R), (u'_G, v'_G), (u'_B, v'_B), (u'_W, v'_W).

В этом случае оценки коэффициентов $\hat{X}_r$, $\hat{Y}_r$, $\hat{Z}_r$, $\hat{X}_g$, $\hat{Y}_g$, $\hat{Z}_g$, $\hat{X}_b$, $\hat{Y}_b$, $\hat{Z}_b$ профиля экрана будут иметь вид:

$$\begin{cases} \begin{bmatrix} \hat{X}_r \\ \hat{X}_g \\ \hat{X}_b \\ \hat{Y}_g \\ \hat{Y}_b \\ \hat{Z}_r \\ \hat{Z}_g \\ \hat{Z}_b \end{bmatrix} = \begin{bmatrix} d'_{11} & d'_{12} & d'_{13} & d'_{14} & d'_{15} & d'_{16} & d'_{17} & d'_{18} \\ d'_{21} & d'_{22} & d'_{23} & d'_{24} & d'_{25} & d'_{26} & d'_{27} & d'_{28} \\ d'_{31} & d'_{32} & d'_{33} & d'_{34} & d'_{35} & d'_{36} & d'_{37} & d'_{38} \\ d'_{41} & d'_{42} & d'_{43} & d'_{44} & d'_{45} & d'_{46} & d'_{47} & d'_{48} \\ d'_{51} & d'_{52} & d'_{53} & d'_{54} & d'_{55} & d'_{56} & d'_{57} & d'_{58} \\ d'_{61} & d'_{62} & d'_{63} & d'_{64} & d'_{65} & d'_{66} & d'_{67} & d'_{68} \\ d'_{71} & d'_{72} & d'_{73} & d'_{74} & d'_{75} & d'_{76} & d'_{77} & d'_{78} \\ d'_{81} & d'_{82} & d'_{83} & d'_{84} & d'_{85} & d'_{86} & d'_{87} & d'_{88} \end{bmatrix}^{-1} \begin{bmatrix} 15u'_1R_1 \\ R_1(9-15v'_1) \\ 15u'_2R_2 \\ R_2(9-15v'_2) \\ 15u'_3R_3 \\ R_3(9-15v'_3) \\ 15u'_4R_4 \\ R_4(9-15v'_4) \end{bmatrix}, \\ \hat{Y}_r = 1 - \hat{Y}_g - \hat{Y}_b \end{cases} \tag{9}$$

где $d'_{11} = R_1(4-u'_1)$, $d'_{12} = G_1(4-u'_1)$, $d'_{13} = B_1(4-u'_1)$, $d'_{14} = 15u'_1(R_1-G_1)$, $d'_{15} = 15u'_1(R_1-B_1)$, $d'_{16} = -3u'_1R_1$, $d'_{17} = -3u'_1G_1$, $d'_{18} = -3u'_1B_1$, $d'_{21} = R_1v'_1$, $d'_{22} = G_1v'_1$, $d'_{23} = B_1v'_1$, $d'_{24} = (R_1-G_1)(9-15v'_1)$, $d'_{25} = (R_1-B_1)(9-15v'_1)$, $d'_{26} = 3R_1v'_1$, $d'_{27} = 3G_1v'_1$, $d'_{28} = 3B_1v'_1$, $d'_{31} = R_2(4-u'_2)$, $d'_{32} = G_2(4-u'_2)$, $d'_{33} = B_2(4-u'_2)$, $d'_{34} = 15u'_2(R_2-G_2)$, $d'_{35} = 15u'_2(R_2-B_2)$, $d'_{36} = -3u'_2R_2$, $d'_{37} = -3u'_2G_2$, $d'_{38} = -3u'_2B_2$, $d'_{41} = R_2v'_2$, $d'_{42} = G_2v'_2$, $d'_{43} = B_2v'_2$, $d'_{44} = (R_2-G_2)(9-15v'_2)$, $d'_{45} = (R_2-B_2)(9-15v'_2)$, $d'_{46} = 3R_2v'_2$, $d'_{47} = 3G_2v'_2$, $d'_{48} = 3B_2v'_2$, $d'_{51} = R_3(4-u'_3)$, $d'_{52} = G_3(4-u'_3)$, $d'_{33} = B_3(4-u'_3)$, $d'_{54} = 15u'_3(R_3-G_3)$, $d'_{55} = 15u'_3(R_3-B_3)$, $d'_{56} = -3u'_3R_3$, $d'_{57} = -3u'_3G_3$, $d'_{58} = -3u'_3B_3$, $d'_{61} = R_3v'_3$, $d'_{62} = G_3v'_3$, $d'_{63} = B_3v'_3$, $d'_{64} = (R_3-G_3)(9-15v'_3)$, $d'_{65} = (R_3-B_3)(9-15v'_3)$, $d'_{66} = 3R_3v'_3$, $d'_{67} = 3G_3v'_3$, $d'_{68} = 3B_3v'_3$, $d'_{71} = R_4(4-u'_4)$, $d'_{72} = G_4(4-u'_4)$, $d'_{73} = B_4(4-u'_4)$, $d'_{74} = 15u'_4(R_4-G_4)$, $d'_{75} = 15u'_4(R_4-B_4)$, $d'_{76} = -3u'_4R_4$, $d'_{77} = -3u'_4G_4$, $d'_{78} = -3u'_4B_4$, $d'_{81} = R_4v'_4$, $d'_{82} = G_4v'_4$, $d'_{83} = B_4v'_4$, $d'_{84} = (R_4-G_4)(9-15v'_4)$, $d'_{85} = (R_4-B_4)(9-15v'_4)$, $d'_{86} = 3R_4v'_4$, $d'_{87} = 3G_4v'_4$, $d'_{88} = 3B_4v'_4$.

Система (9) получена путем решения системы уравнений

$$\begin{cases} 4\left(\hat{X}_rR_1 + \hat{X}_gG_1 + \hat{X}_bB_1\right) = u'_1\left(\left(\hat{X}_r + 15\hat{Y}_r + 3\hat{Z}_r\right)R_1 + \left(\hat{X}_g + 15\hat{Y}_g + 3\hat{Z}_g\right)G_1 + \left(\hat{X}_b + 15\hat{Y}_b + 3\hat{Z}_b\right)B_1\right), \\ 9\left(\hat{Y}_rR_1 + \hat{Y}_gG_1 + \hat{Y}_bB_1\right) = v'_1\left(\left(\hat{X}_r + 15\hat{Y}_r + 3\hat{Z}_r\right)R_1 + \left(\hat{X}_g + 15\hat{Y}_g + 3\hat{Z}_g\right)G_1 + \left(\hat{X}_b + 15\hat{Y}_b + 3\hat{Z}_b\right)B_1\right), \\ 4\left(\hat{X}_rR_2 + \hat{X}_gG_2 + \hat{X}_bB_2\right) = u'_2\left(\left(\hat{X}_r + 15\hat{Y}_r + 3\hat{Z}_r\right)R_2 + \left(\hat{X}_g + 15\hat{Y}_b + 3\hat{Z}_g\right)G_2 + \left(\hat{X}_b + 15\hat{Y}_b + 3\hat{Z}_b\right)B_2\right), \\ 9\left(\hat{Y}_rR_2 + \hat{Y}_gG_2 + \hat{Y}_bB_2\right) = v'_2\left(\left(\hat{X}_r + 15\hat{Y}_r + 3\hat{Z}_r\right)R_2 + \left(\hat{X}_g + 15\hat{Y}_g + 3\hat{Z}_g\right)G_2 + \left(\hat{X}_b + 15\hat{Y}_b + 3\hat{Z}_b\right)B_2\right), \\ 4\left(\hat{X}_rR_3 + \hat{X}_gG_3 + \hat{X}_bB_3\right) = u'_3\left(\left(\hat{X}_r + 15\hat{Y}_r + 3\hat{Z}_r\right)R_3 + \left(\hat{X}_g + 15\hat{Y}_g + 3\hat{Z}_g\right)G_3 + \left(\hat{X}_b + 15\hat{Y}_b + 3\hat{Z}_b\right)B_3\right), \\ 9\left(\hat{Y}_rR_3 + \hat{Y}_gG_3 + \hat{Y}_bB_3\right) = v'_3\left(\left(\hat{X}_r + 15\hat{Y}_r + 3\hat{Z}_r\right)R_3 + \left(\hat{X}_g + 15\hat{Y}_g + 3\hat{Z}_g\right)G_3 + \left(\hat{X}_b + 15\hat{Y}_b + 3\hat{Z}_b\right)B_3\right), \\ 4\left(\hat{X}_rR_4 + \hat{X}_gG_4 + \hat{X}_bB_4\right) = u'_4\left(\left(\hat{X}_r + 15\hat{Y}_r + 3\hat{Z}_r\right)R_4 + \left(\hat{X}_g + 15\hat{Y}_g + 3\hat{Z}_g\right)G_4 + \left(\hat{X}_b + 15\hat{Y}_b + 3\hat{Z}_b\right)B_4\right), \\ 9\left(\hat{Y}_rR_4 + \hat{Y}_gG_4 + \hat{Y}_bB_4\right) = v'_4\left(\left(\hat{X}_r + 15\hat{Y}_r + 3\hat{Z}_r\right)R_4 + \left(\hat{X}_g + 15\hat{Y}_g + 3\hat{Z}_g\right)G_4 + \left(\hat{X}_b + 15\hat{Y}_b + 3\hat{Z}_b\right)B_4\right), \\ \hat{Y}_r = 1 - \hat{Y}_g - \hat{Y}_b. \end{cases}$$

6. Оценка коэффициентов профиля экрана по результатам измерений, полученным в параметрах цветовой модели *XYZ*

Существующие разновидности контрольно-измерительных приборов, используемых в колориметрических экспериментах, позволяют получать результаты измерения цветовых координат в параметрах цветовой модели *XYZ*.

В этом случае для оценки коэффициентов $\widehat{X}_r$, $\widehat{Y}_r$, $\widehat{Z}_r$, $\widehat{X}_g$, $\widehat{Y}_g$, $\widehat{Z}_g$, $\widehat{X}_b$, $\widehat{Y}_b$, $\widehat{Z}_b$ профиля экрана необходимо отобразить на экране произвольным образом заданные кодами (R_1, G_1, B_1), (R_2, G_2, B_2), (R_3, G_3, B_3) три цвета и провести измерения соответствующих им координат цвета (X_1, Y_1, Z_1), (X_2, Y_2, Z_2), (X_3, Y_3, Z_3) [12].

Оценки коэффициентов $\widehat{X}_r$, $\widehat{Y}_r$, $\widehat{Z}_r$, $\widehat{X}_g$, $\widehat{Y}_g$, $\widehat{Z}_g$, $\widehat{X}_b$, $\widehat{Y}_b$, $\widehat{Z}_b$ профиля экрана будут иметь вид

$$\begin{bmatrix} \widehat{X}_r \\ \widehat{X}_g \\ \widehat{X}_b \\ \widehat{Y}_r \\ \widehat{Y}_g \\ \widehat{Y}_b \\ \widehat{Z}_r \\ \widehat{Z}_g \\ \widehat{Z}_b \end{bmatrix} = \begin{bmatrix} R_1 & G_1 & B_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & R_1 & G_1 & B_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & R_1 & G_1 & B_1 \\ R_2 & G_2 & B_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & R_2 & G_2 & B_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & R_2 & G_2 & B_2 \\ R_3 & G_3 & B_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & R_3 & G_3 & B_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & R_3 & G_3 & B_3 \end{bmatrix}^{-1} \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ X_2 \\ Y_2 \\ Z_2 \\ X_3 \\ Y_3 \\ Z_3 \end{bmatrix}. \quad (10)$$

Уравнение (10) получено путем решения системы уравнений, основанных на базовом преобразовании (1):

$$\begin{cases} \widehat{X}_r R_1 + \widehat{X}_g G_1 + \widehat{X}_b B_1 = X_1 \\ \widehat{Y}_r R_1 + \widehat{Y}_g G_1 + \widehat{Y}_b B_1 = Y_1 \\ \widehat{Z}_r R_1 + \widehat{Z}_g G_1 + \widehat{Z}_b B_1 = Z_1 \\ \widehat{X}_r R_2 + \widehat{X}_g G_2 + \widehat{X}_b B_2 = X_2 \\ \widehat{Y}_r R_2 + \widehat{Y}_g G_2 + \widehat{Y}_b B_2 = Y_2 \\ \widehat{Z}_r R_2 + \widehat{Z}_g G_2 + \widehat{Z}_b B_2 = Z_2 \\ \widehat{X}_r R_3 + \widehat{X}_g G_3 + \widehat{X}_b B_3 = X_3 \\ \widehat{Y}_r R_3 + \widehat{Y}_g G_3 + \widehat{Y}_b B_3 = Y_3 \\ \widehat{Z}_r R_3 + \widehat{Z}_g G_3 + \widehat{Z}_b B_3 = Z_3 \end{cases}.$$

7. Оценка коэффициентов профиля экрана по результатам измерений, полученным в параметрах цветовой модели *Yxy*

Существующие разновидности контрольно-измерительных приборов, используемых в колориметрических экспериментах, позволяют получать результаты измерения цветовых координат в параметрах цветовой модели *Yxy*.

В этом случае для оценки коэффициентов $\widehat{X}_r$, $\widehat{Y}_r$, $\widehat{Z}_r$, $\widehat{X}_g$, $\widehat{Y}_g$, $\widehat{Z}_g$, $\widehat{X}_b$, $\widehat{Y}_b$, $\widehat{Z}_b$ профиля экрана необходимо отобразить на экране произвольным образом заданные кодами (R_1, G_1, B_1), (R_2, G_2, B_2), (R_3, G_3, B_3) три цвета и провести измерения соответствующих им колориметрических параметров (Y_1, x_1, y_1), (Y_2, x_2, y_2), (Y_3, x_3, y_3).

Оценки коэффициентов $\widehat{X}_r,\ \widehat{Y}_r,\ \widehat{Z}_r,\ \widehat{X}_g,\ \widehat{Y}_g,\ \widehat{Z}_g,\ \widehat{X}_b,\ \widehat{Y}_b,\ \widehat{Z}_b$ матрицы профиля экрана будут иметь вид

$$\begin{bmatrix}\widehat{X}_r\\ \widehat{X}_g\\ \widehat{X}_b\\ \widehat{Y}_r\\ \widehat{Y}_g\\ \widehat{Y}_b\\ \widehat{Z}_r\\ \widehat{Z}_g\\ \widehat{Z}_b\end{bmatrix} = \begin{bmatrix} R_1 & G_1 & B_1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & R_1 & G_1 & B_1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & R_1 & G_1 & B_1\\ R_2 & G_2 & B_2 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & R_2 & G_2 & B_2 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & R_2 & G_2 & B_2\\ R_3 & G_3 & B_3 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & R_3 & G_3 & B_3 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & R_3 & G_3 & B_3\end{bmatrix}^{-1} \begin{bmatrix} Y_1 x_1/y_1\\ Y_1\\ Y_1(1-x_1-y_1)/y_1\\ Y_2 x_2/y_2\\ Y_2\\ Y_2(1-x_2-y_2)/y_2\\ Y_3 x_3/y_3\\ Y_3\\ Y_3(1-x_3-y_3)/y_3\end{bmatrix}. \quad (11)$$

Уравнение (11) получено путем подстановки выражений (2) в (10).

8. Оценка коэффициентов профиля экрана по результатам измерений, полученным в параметрах цветовой модели *Yu′v′*

Существующие разновидности контрольно-измерительных приборов, используемых в колориметрических экспериментах, позволяют получать результаты измерения цветовых координат в параметрах цветовой модели *Yu′v′*.

В этом случае для оценки коэффициентов $\widehat{X}_r,\ \widehat{Y}_r,\ \widehat{Z}_r,\ \widehat{X}_g,\ \widehat{Y}_g,\ \widehat{Z}_g,\ \widehat{X}_b,\ \widehat{Y}_b,\ \widehat{Z}_b$ профиля экрана необходимо отобразить на экране произвольным образом заданные кодами (R_1, G_1, B_1), (R_2, G_2, B_2), (R_3, G_3, B_3) три цвета и провести измерения соответствующих им колориметрических параметров (Y_1, u'_1, v'_1), (Y_2, u'_2, v'_2), (Y_3, u'_3, v'_3).

Оценки коэффициентов $\widehat{X}_r,\ \widehat{Y}_r,\ \widehat{Z}_r,\ \widehat{X}_g,\ \widehat{Y}_g,\ \widehat{Z}_g,\ \widehat{X}_b,\ \widehat{Y}_b,\ \widehat{Z}_b$ профиля экрана будут иметь вид:

$$\begin{bmatrix}\widehat{X}_r\\ \widehat{X}_g\\ \widehat{X}_b\\ \widehat{Y}_r\\ \widehat{Y}_g\\ \widehat{Y}_b\\ \widehat{Z}_r\\ \widehat{Z}_g\\ \widehat{Z}_b\end{bmatrix} = \begin{bmatrix} R_1 & G_1 & B_1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & R_1 & G_1 & B_1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & R_1 & G_1 & B_1\\ R_2 & G_2 & B_2 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & R_2 & G_2 & B_2 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & R_2 & G_2 & B_2\\ R_3 & G_3 & B_3 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & R_3 & G_3 & B_3 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & R_3 & G_3 & B_3\end{bmatrix}^{-1} \begin{bmatrix} 9Y_1 u'_1/4v'_1\\ Y_1\\ (3Y_1(4-u'_1)/4v'_1) - 5Y_1\\ 9Y_2 u'_2/4v'_2\\ Y_2\\ (3Y_2(4-u'_2)/4v'_2) - 5Y_2\\ 9Y_3 u'_3/4v'_3\\ Y_3\\ (3Y_3(4-u'_3)/4v'_3) - 5Y_3\end{bmatrix}. \quad (12)$$

Уравнение (12) получено путем подстановки выражений (2) в (10).

Заключение

Выбор способа вычисления оценок коэффициентов матрицы профиля экранов определяется возможностями инструментальных средств, доступных в колориметрических экспериментах, и наличием в технической документации на экраны цветовых координат, определяющих положение треугольника цветового охвата экранов на цветовой плоскости [13, 14].

Рабочими плоскостями являются цветовые плоскости *xy* и *u′v′*. Инструментальные средства, необходимые для оценки коэффициентов матрицы профиля экранов, должны обеспечивать измерения в параметрах хотя бы в одной из стандартизованных цветовых моделей: *XYZ*, *Yxy*, *Yu′v′*.

При решении задачи выбора исследователям необходимо учитывать следующее:

- оценки коэффициентов $\widehat{X}_r,\ \widehat{Y}_r,\ \widehat{Z}_r,\ \widehat{X}_g,\ \widehat{Y}_g,\ \widehat{Z}_g,\ \widehat{X}_b,\ \widehat{Y}_b,\ \widehat{Z}_b$, полученные по типовым значениям координат цветности вершин треугольника цветового охвата определенной модели экрана,

в общем случае не соответствуют свойствам цветовоспроизведения каждого конкретного образца экрана вследствие технологического разброса, присущего изготовлению технических средств на основе полупроводников;

- измерение цветовых координат экранов в параметрах рассмотренных цветовых моделей осуществляется инструментальными средствами с погрешностью, определяемой условиями измерения (уровень и спектральный состав внешнего освещения) и характеристиками измерительной аппаратуры;
- оценки коэффициентов матрицы профиля экрана, полученные расчетным путем как на основе колориметрических измерений, так и на основе обработки данных технической документации экранов, необходимо проверять на физическую реализуемость (матрица профиля экрана (1) должна иметь обратную матрицу).

Вычислительная сложность математического аппарата, применяемого для всех предложенных способов оценки коэффициентов матрицы профиля экранов, примерно одинаковая.

Список литературы

1. **Ibraheem N. A., Hasan M. M., Khan R. Z., Mishra P. K.** Understanding color models: a review // ARPN Journal of science and technology. 2012. Vol. 2, N. 3. P. 265—275.
2. **Schanda J.** Colorimetry Understanding the CIE System. Wiley-Interscience John Wiley & Sons, INC. Publication, 2007. 499 p.
3. **Evanicky D., Granger Ed., Ingulsrud J., Meng A. T.** US Patent 2009/0051711 A1: "Compact flat panel color calibration system", Feb. 26. 2009.
4. **Kwak Y., Lee S., Choe W., Kim Ch.-Y.** Optimal chromaticities of the primaries for gamut 3-channel display// Proceeding of SPIE-IS&T Electronic Imaging. 2005. Vol. 5667. P. 319—327.
5. **Mang O.-Y., Huang T.-W., Hsieh Y.-F., Kuob Y.-T.** Research of the chromaticity coordinates and color spectrum calibration using tristimulus sensors and eigenspectrum method // Proceeding of SPIE "Optical inspection and metrology for non-optics industries". 2009. Vol. 7432. Art. 743214.
6. **Rodrıguez-Pardo C. E., Gaurav Sh., Feng X.-F., Speigle J., Sezan I.** Optimal gamut volume design for three primary and multiprimary display systems // Proceeding of SPIE-IS&T Electronic Imaging. 2012. Vol. 8292. Art. 82920C.
7. **Rolkosky D. J., Dagnelie G., Kramer K., Havey G., Seifert G. J.** Calibration tools for PC-based vision assessment// Proceeding of IEEE Eng. Med. Biol. Soc. 2009. P. 781—784.
8. **Zargaryants G. S., Mikhailov O. M.** Integral remote colorimeter bases on the RGB colorimetric system // Light & Engineering. 2008. Vol. 16, N. 3. P. 69—77.
9. **Жаринов И. О., Жаринов О. О**. Исследование математической модели цветопередачи жидкокристаллических панелей // Программная инженерия. 2015. № 5. С. 32—42.
10. **Жаринов И. О., Жаринов О. О.** Оценка параметров математической модели цветопередачи жидкокристаллической панели // Информационно-управляющие системы. 2015. № 2. С. 49—56.
11. **Жаринов И. О., Жаринов О. О.** Математическое обеспечение для решения практической задачи калибровки мониторов на жидких кристаллах и светодиодах // Программная инженерия. 2016. Т. 7, № 4. С. 173—180.
12. **Жаринов И. О., Жаринов О. О.** Метод программной компенсации технологического разброса координат цветности жидкокристаллических панелей // Научно-технический вестник информационных технологий, механики и оптики, 2015. Т. 15, № 3. С. 387—397.
13. **Gatchin Y. A., Zharinov I. O., Korobeynikov A. G., Zharinov O. O.** Theoretical estimation of Grassmann's transformation resolution in avionics color coding systems // Modern Applied Science. 2015. Vol. 9, N. 5. P. 197—210.
14. **Aleksanin S. A., Zharinov I. O., Korobeynikov A. G., Perezyabov O. A., Zharinov O. O.** Evaluation of chromaticity coordinate shifts for visually perceived image in terms of exposure to external illuminance //ARPN Journal of Engineering and Applied Sciences. 2015. Vol. 10, N. 17. P. 7494—7501.

Methods of Evaluation of Coefficients of Color Profile Matrix for Displays with Three Primary Color Reproduction System

I. O. Zharinov, igor_rabota@pisem.net, Saint Petersburg National Research University of Information Technologies, Mechanics and Optics (ITMO University), Saint Petersburg, 197101, Russian Federation, Design Bureau "Electroavtomatika", Saint Petersburg, 198095, Russian Federation,
O. O. Zharinov, zharinov73@hotbox.ru, Saint-Petersburg State University of Aerospace Instrumentation, Saint Petersburg, 190000, Russian Federation

Corresponding author:

Zharinov Igor O., Chief of Department, Saint Petersburg National Research University of Information Technologies, Mechanics and Optics (ITMO University), 197101, Saint Petersburg, Russian Federation
e-mail: igor_rabota@pisem.net

Received on April 22, 2016
Accepted on May 31, 2016

The problem of evaluation of coefficients of color profile matrix for displays with three primary colors reproduction system is considered. According to three primary colors reproduction system, color of each pixel is created as combination of three sources of basic colors: red, green and blue. Such sources may be light emitting diodes or liquid crystals that polarize white backlight. Color profile matrix determines transformation of RGB codes into XYZ

color coordinates and color gamut of thedisplay depends upon matrix coefficients. Display manufacturers tend not to include values of coefficients of display's profile matrix in corresponding technical manuals, whilst these data are necessary for advanced color management procedures. In this article seven different methods of evaluation of coefficients of display's color profile matrix are proposed. To build all those methods, two approaches were used. First approach is based upon data in display's technical manual, such as chromaticity coordinates of vertices of gamut triangle and white point chromaticity coordinates. Second approach is developed assuming results of experimental data, which may be obtained during colorimetric measurements in different color spaces. Measuring instruments to be used in colorimetric measurements must provide measurements at least in one of standard color spaces, such as XYZ, Yxy, Yu'v'.

Keywords: *color spaces, color profile matrix, display, colorimetric measurements, color coordinates, chromaticity coordinates*

For citation:

Zharinov I. O., Zharinov O. O. Methods of Evaluation of Coefficients of Color Profile Matrix for Displays with Three Primary Color Reproduction System, *Programmnaya Ingeneria*, 2016, vol. 7, no. 9, pp. 414—423.

DOI: 10.17587/prin.7.414-423

References

1. **Ibraheem N. A., Hasan M. M., Khan R. Z., Mishra P. K.** Understanding color models: a review, *ARPN Journal of science and technology*, 2012, vol. 2, no. 3, pp. 265—275.

2. **Schanda J.** *Colorimetry Understanding the CIE System,* Wiley-Interscience John Wiley & Sons, INC., Publication, 2007, 499 p.

3. **Evanicky D., Granger Ed., Ingulsrud J., Meng A. T.** US Patent 2009/0051711 A1: "Compact flat panel color calibration system", Feb. 26, 2009.

4. **Kwak Y., Lee S., Choe W., Kim Ch.-Y.** Optimal chromaticities of the primaries for gamut 3-channel display, *Proceeding of SPIE-IS&T Electronic Imaging*, 2005, vol. 5667, pp. 319—327.

5. **Mang O.-Y., Huang T.-W., Hsieh Y.-F., Kuob Y.-T.** Research of the chromaticity coordinates and color spectrum calibration using tristimulus sensors and eigenspectrum method, *Proceeding of SPIE "Optical inspection and metrology for non-optics industries"*, 2009, vol. 7432, Art. 743214.

6. **Rodrıguez-Pardo C. E., Gaurav Sh., Feng X.-F., Speigle J., Sezan I.** Optimal gamut volume design for three primary and multiprimary display systems, *Proceeding of SPIE-IS&T Electronic Imaging*, 2012, vol. 8292, Art. 82920C.

7. **Rolkosky D. J., Dagnelie G., Kramer K., Havey G., Seifert G. J.** Calibration tools for PC-based vision assessment, *Proceeding of IEEE Eng. Med. Biol. Soc.*, 2009, pp. 781—784.

8. **Zargaryants G. S., Mikhailov O. M.** Integral remote colorimeter bases on the RGB colorimetric system, *Light & Engineering*, 2008, vol. 16, no. 3, pp. 69—77.

9. **Zharinov I. O., Zharinov O. O.** Issledovanie matematicheskoj modeli cvetoperedachi zhidkokristallicheskih panelej (The research of the mathematical model of color representation of LCD-panels), *Programmnaya Ingeneria*, 2015, no. 5, pp. 32—42 (in Russian).

10. **Zharinov I. O., Zharinov O. O.** Ocenka parametrov matematicheskoj modeli cvetoperedachi zhidkokristallicheskoj paneli (Evaluation of parameters of the mathematical model of color response performances of LCD-panel), *Informacionno-Upravljajushhie Sistemy*, 2015, no. 2, pp. 49—56 (in Russian).

11. **Zharinov I. O., Zharinov O. O.** Matematicheskoe obespechenie dlja reshenija prakticheskoj zadachi kalibrovki monitorov na zhidkih kristallah i svetodiodah (Software for solving the practical problem of calibration liquid crystal displays and light-emitting diodes), *Programmnaya ingeneria*, 2016, vol. 7, no. 4, pp. 173—180 (in Russian).

12. **Zharinov I. O., Zharinov O. O.** Metod programmnoj kompensacii tehnologicheskogo razbrosa koordinat cvetnosti zhidkokristallicheskih panelej (Method of software-based compensation of technological variation in chromaticity coordinates of LCD panels), *Nauchno-Tehnicheskii Vestnik Informatsionnykh Tekhnologii, Mekhaniki i Optiki*, 2015, vol. 15, no. 3, pp. 387—397 (in Russian).

13. **Gatchin Y. A., Zharinov I. O., Korobeynikov A. G., Zharinov O. O.** Theoretical estimation of Grassmann's transformation resolution in avionics color coding systems, *Modern Applied Science*, 2015, vol. 9, no. 5, pp. 197—210.

14. **Aleksanin S. A., Zharinov I. O., Korobeynikov A. G., Perezyabov O. A., Zharinov O. O.** Evaluation of chromaticity coordinate shifts for visually perceived image in terms of exposure to external illuminance, *ARPN Journal of Engineering and Applied Sciences*, 2015, vol. 10, no. 17, pp. 7494—7501.

ИНФОРМАЦИЯ

УДК 004.82 DOI: 10.17587/prin.7.424-431

К. И. Костенко, канд. физ.-мат. наук, зав. каф., e-mail kostenko@kubsu.ru,
Кубанский государственный университет, г. Краснодар

Моделирование оператора вывода для иерархических формализмов знаний

Иерархические формализмы представления знаний являются эффективным инструментом построения моделей содержания конкретных предметных областей. Практическое применение таких моделей в профессиональной деятельности нуждается в формализации понятия механизма вывода, связанного с реализацией процессов построения решений отдельных задач. Для этого используют инварианты структурного представления задачи и знания, синтезируемого в целях нахождения решения, основанные на конструкциях шаблона и знания иерархической структуры. Операторы вывода реализуются процессами синтеза знаний, определяемыми системами специальных правил. Решения задач извлекают из синтезированных знаний с помощью отображений трассирования. Приведены элементы универсального языка конструирования постановок задач, сценариев и правил для класса иерархических формализмов представления знаний, включающего элементы инженерии задач и процессов их решения.

Ключевые слова: *формализм знаний, задача, декомпозиция задачи, оператор вывода, синтез знаний, правило синтеза*

Введение

Оператор вывода относится к стандартным элементам структурно-функциональной модели любой интеллектуальной системы. Универсальное понятие оператора вывода предполагает уточнение его свойств, согласованное с атрибутами формализма представления знаний. Для этого достаточно дополнить концепцию формализма знаний инвариантами областей определения и значений оператора вывода, а также инвариантом декомпозиции задач в схемы реализации и извлечения результатов вывода. С учетом сделанных замечаний всякий оператор вывода может быть формализован как отображение, которое моделирует процессы обработки знаний. Такие процессы соответствуют понятию вывода для дедуктивных систем. Структурированное описание оператора вывода связано с определением системы этапов указанных процессов, реализующих конструирование решений задач по формальным постановкам.

Формализмы представления знаний

Формализмы представления знаний составляют специальный класс математических систем, применяемых для теоретического исследования свойств знаний и операций над знаниями, позволяющих моделировать процессы человеческого мышления. Универсальная концепция такого формализма реализуется через построение системы унифицированных инвариантов, отражающих содержательные представления о фундаментальных атрибутах знаний. Отличие формализма знаний от модели отдельной области знаний состоит в том, что всякая модель оперирует конкретным множеством объектов, отражающим содержание области знаний, и, как правило, является частью семейства объектов применяемого формализма знаний.

Уточнение общих структурных атрибутов произвольных формализмов знаний реализуется определением, использующим инварианты операции композиции и отношения вложения на множестве фрагментов знаний. С рассматриваемых позиций всякий формализм знаний задается как четверка $\Im = (M, D, \circ, \leqslant)$. Здесь D — перечислимое множество фрагментов знаний; M — разрешимое подмножество множества D, составленное представлениями отдельных знаний; $\circ$ — операция композиции $(\circ : D \times D \to D)$; $\leqslant$ — отношения вложения на D. Дополнительное общее требование для всякого формализма знаний связано с существованием "пустого" знания $\Lambda \in M$, для которого $\forall z \in M(\Lambda \leqslant z)$. Прикладной аспект Λ связан с обеспечением замкнутости операции композиции на множествах фрагментов знаний, имеющих смысл для конкретной области знаний, когда некоторые композиции фрагментов знаний полагаются равными пустому знанию, если они не имеют смысла в указанной области.

Элементарные и простые знания

Универсальным требованием к операции композиции является возможность составления произвольных фрагментов знаний в рассматриваемом формализме в виде композиции элементарных и простых знаний. Концепция элементарного знания, используемая в настоящей работе, близка подходу Фреге к моделированию семантики формальных объектов [1]. Она заключается в структурной неделимости каждого элементарного знания. Содержание каждого такого знания может быть сложным и раскрываться семантическими структурами, составленными из элементарных знаний и семантических отношений между фрагментами структур. Структуры, представляющие содержание элементарных знаний, являются элементами областей определения и значения морфизмов, моделирующих операции обработки знаний. Элементарными знаниями для применяемого далее формализма абстрактного пространства знаний являются элементарные конфигурации таких пространств и семантические отношения, используемые для конструирования сложных конфигураций [2].

Всякое простое знание образуют два элементарных знания, связанных семантическим отношением. Базу знаний конкретной области деятельности составляют множества элементарных и простых знаний, извлекаемые операциями анализа содержания этой области. Сложные знания составляются из элементов базы знаний с помощью процессов, реализующих операторы вывода. Управление такими процессами осуществляется с помощью правил построения сложных знаний, из которых извлекаются решения задач. Поэтому область определения каждого оператора вывода составляют перечислимые множества элементарных и простых знаний, а также постановок задач, согласованных с форматами используемого формализма знаний.

Инвариант постановки задачи

Всякая задача является логической комбинацией простых задач. Универсальный формат постановки простой задачи для произвольного формализма знаний образуют композиции элементарных и неизвестных фрагментов знаний. Эти композиции определяют алгебраическую структуру решения, извлекаемого из синтезируемых оператором вывода композиций элементарных знаний. Для этого применяют отображения трассирования иерархических структур [2]. Элементы постановок задач, обозначаемые символами неизвестных, называют целями. Они соответствуют фрагментам знаний, устанавливаемых с помощью операторов вывода. Структурное представление постановки простой задачи имеет вид бинарного дерева. Внутренние вершины этого дерева размечены операцией композиции, а висячие вершины — элементарными знаниями и обозначениями неизвестных, указывающими на цели задачи.

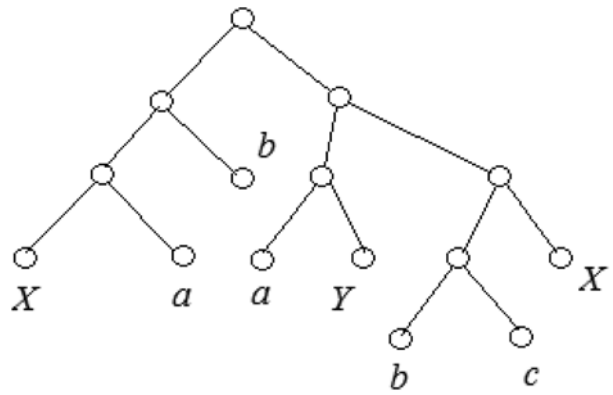


Рис. 1. Дерево постановки задачи
$((X \circ a) \circ b) \circ ((a \circ Y) \circ ((b \circ c) \circ X))$

Примером постановки простой задачи служит выражение: $\big((X \circ a) \circ b\big) \circ \big((a \circ Y) \circ \big((b \circ c) \circ X\big)\big)$. Здесь X и Y — цели заданной задачи, а a, b, c — конкретные элементарные знания. Дерево, соответствующее рассматриваемой задаче, приведено на рис. 1. При этом операция композиции, используемая для разметки внутренних вершин дерева, на рисунке не указывается.

Фрагменты синтезированного для решаемой задачи сложного знания, соответствующие одинаково обозначаемым неизвестным, должны совпадать.

Структуры бинарных деревьев для унифицированных постановок задач, относящихся к произвольным формализмам знаний, отличаются от структур представлений конфигураций для формализма абстрактного пространства знаний [2]. Отличие состоит в том, что разметкой внутренних вершин в дереве постановки задачи является операция композиции, а разметку внутренних вершин структур конфигураций составляют семантические отношения, которые выполняются между конфигурациями, представляемыми левым и правым поддеревьями таких вершин. Для абстрактных пространств знаний удобно представление задач, согласованное со структурами конфигураций, в которых внутренние вершины деревьев размечены семантическими отношениями, а висячие — элементарными конфигурациями и символами переменных, обозначающими конфигурации, являющиеся целями задач. На рис. 2 изображены деревья, представляющие простые задачи "*дом имеет_частью X*" и "*X имеет_свойство устойчив к нагреванию*", в которых *имеет_частью* (обозначается как $\lhd$) и *имеет_свойство* $\left(\stackrel{\Delta}{=}\right)$ — это элементарные

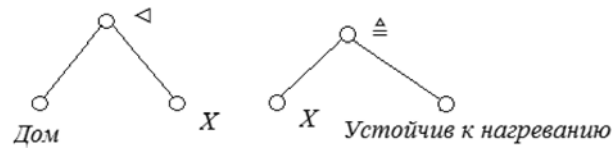


Рис. 2. Деревья представления простых задач для абстрактного пространства знаний

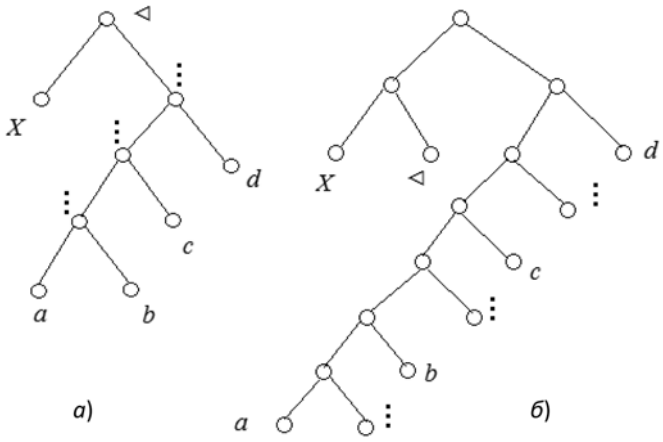


Рис. 3. Трансформация дерева задачи для формализма абстрактного пространства знаний в общий формат задачи

знания, представляющие конкретные семантические отношения.

Существует обратимая трансформация постановок задач к базам знаний, основанных на формализме абстрактного пространства знаний, в универсальную структуру постановки задачи для произвольных формализмов представления знаний [4].

Рассмотрим пример такой трансформации для задачи, приведенной на рис. 3, *а*. Целью указанной задачи является элементарное знание, обозначающее понятие, которое соответствует объектам, связанным отношением агрегирования с элементами четырех классов объектов, представленных понятиями *a*, *b*, *c*, *d*. Здесь символ ⋮ обозначает вспомогательное отношение формирования параллельной серии [3]. На рис. 3, *б*, представлено дерево постановки той же задачи в универсальном формате, задаваемое композицией

$$(X \circ \triangleleft) \circ \Big(\big(\big(\big(\big((a \circ \vdots) \circ b\big) \circ \vdots\big) \circ c\big) \circ \vdots\big) \circ d\Big).$$

Понятие постановки задачи является общим и абстрактным. Ему соответствует универсальный механизм решения задач, представленный оператором вывода. Для произвольных формализмов знаний допустим алгоритм реализации оператора вывода, основанный на схеме перебора вариантов. Если постановка задачи T содержит символы переменных x_1, ..., x_n, то процесс решения такой задачи связан с построением всех подстановок $\Theta = \left(\frac{x_1}{t_1}, \ldots, \frac{x_n}{t_n}\right)$, где t_1, ..., t_n — элементы множества D используемого формализма знаний. Для каждой такой подстановки Θ конструируется выражение $T\Theta$, получаемое из T заменой всех вхождений x_1, ..., x_n на t_1, ..., t_n. Если в полученном выражении применение всякой операции композиции не равно Λ, то Θ объявляется решением T.

Рассмотрим класс канонических задач, структура которого содержит ровно одну вершину, размеченную отношением. Простая классификация таких задач связана с положением символов неизвестных относительно корня бинарного дерева. Пусть a и b — конкретные значения, а X и Y — символы неизвестных. Тогда задача $a\rho b$ называется проверкой, $a\rho X$ — прямой задачей (неизвестная справа), $X\rho b$ — обратной задачей (неизвестная слева), а $X\rho Y$ — открытой задачей (неизвестные слева и справа). Решения приведенных задач основываются на обработке множеств элементарных и простых знаний, составляющих формализованную модель области знаний.

Решение задачи любого из перечисленных типов сводится к построению всех корректных в заданной области знаний пар фрагментов знаний, принадлежащих отношению ρ. В приложениях можно ограничиться использованием специальных систем правил синтеза сложных структур знаний из элементов базы знаний моделируемой предметной области знаний. Для этого из конструируемого множества извлекают пары с заданными знаниями, принадлежащие отношению ρ (проверка), заданным знанием *a* слева или справа в паре из ρ (прямая или обратная задача), а также все пары, принадлежность которых отношению ρ следует из содержания моделируемой области знаний. Отметим, что решение канонической задачи не может быть получено только из описания отношения ρ для формализма представления знаний. Последнее вызвано отмеченной особенностью формализованных моделей областей знаний, являющихся сужениями формализмов представления знаний.

Формат постановки простой задачи может не совпадать с форматами для канонических задач. Например, на рис. 1 приведена неканоническая задача. Возможны разные способы трансформации неканонических задач в канонические. Один из них связан с заменой заданной неканонической задачи на комбинацию канонических задач, из решений которых составляется решение исходной задачи. Для этого применяют специальные соотношения, которые должны выполняться для решений канонических задач, составляющих указанные комбинации. Например, задача, приведенная на рис. 4, *а*, может быть сведена к двум задачам, приведенным на рис. 4, *б*.

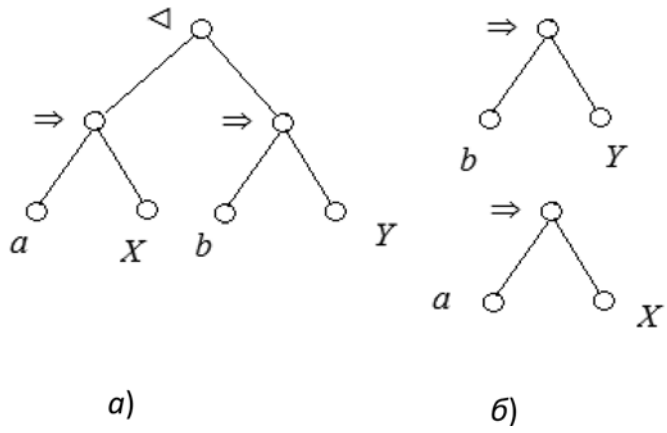


Рис. 4. Сведение неканонической задачи к каноническим задачам

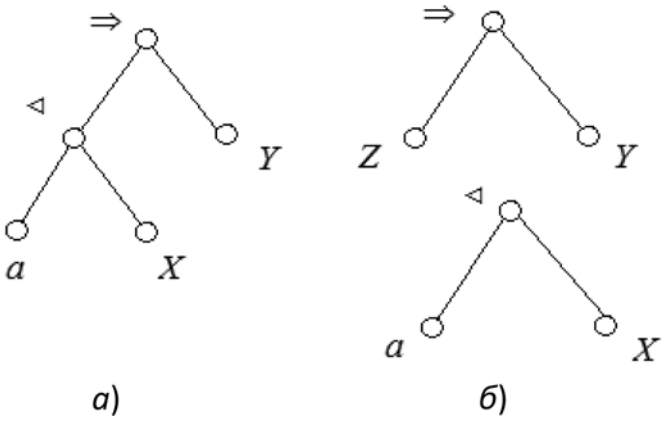


Рис. 5. Трансформация неканонической задачи в канонические задачи

На рис. 4 символ $\Rightarrow$ обозначает отношение дедуктивной выводимости между формулами, задающими свойства объектов области знаний. Задача, представленная на рис. 4, *а*, связана с нахождением объектов, обозначаемых как X и Y, для которых выполняется соотношение "$b \Rightarrow Y$ — это часть утверждения о дедуктивной выводимости $a \Rightarrow X$". Последнее означает, что $a \triangleleft (b \Rightarrow Y)$ или $X \triangleleft (b \Rightarrow Y)$. Решения канонических задач $a \Rightarrow X$ и $b \Rightarrow Y$ являются формулами, выводимыми из a и b. Механизм вывода конструирует решения неканонической задачи из решений задач $a \Rightarrow X$ и $b \Rightarrow Y$. Для этого для каждой пары решений канонических задач c и d (являющихся формулами, выводимыми из a и b), дополнительно проверяется справедливость отношения $(a \Rightarrow c) \triangleleft (b \Rightarrow d)$.

Рассмотренный пример представляет общую схему нахождения решений неканонических простых задач, в которой дополнительно проверяется допустимость замены символов неизвестных на решения простых задач, а также выполнимость всех отношений в структуре простой задачи для результата такой замены. Аналогичным образом трансформируется неканоническая задача, изображенная на рис. 5, *а*. Ей соответствуют канонические задачи, представленные на рис. 5, *б*.

В рассматриваемом случае применена дополнительная неизвестная Z, и каждое решение канонической задачи $Z \Rightarrow Y$ должно удовлетворять условию совпадения значения Z и фрагмента $a \triangleleft b$, где b является решением канонической задачи $a \triangleleft X$.

Управление оператором вывода для пространств знаний

Оператор вывода — это функциональный инвариант для концепции формализма знаний, связанный с процессами решения задач с помощью перечислимых множеств знаний применяемого формализма. Оператору вывода соответствует отображение $\Psi: T \times K \to D^*$, порождающее указанные процессы в форме вычислимых последовательности фрагментов представлений знаний. Область определения Ψ составляют элементы множества $T \times K$, где T — множество всех постановок задач, а K — семейство перечислимых подмножеств множества D. Значением $\Psi(t, k)$ является вычислимая последовательность фрагментов знаний $\Psi(t, k) \in D^*$. Решение конкретной задачи $t \in T$ по $k \in K$, где $k \in K$ — заданное состояние базы знаний, извлекается из $\Psi(t, k)$.

Реализация оператора вывода определяет способ нахождения решений задач, относящийся к некоторому классу формализмов знаний. Рассматриваемая далее универсальная адаптация понятия оператора вывода для формализма знаний опирается на инварианты декомпозиции задачи и правила синтеза фрагментов знаний. При этом декомпозиция канонической задачи связана с уточнением структуры задачи, представляемой как результат агрегирования подзадач.

Правила решения канонических задач

Правилами реализуются схемы синтеза сложных знаний. Множество правил составляют классы, относящиеся к конкретным типам задач. Для моделирования правил синтеза сложных знаний далее будем использовать формат конфигураций абстрактных пространств знаний, конструируемых из элементарных конфигураций. Уточним элементы описаний структур конфигураций. Пусть I — множество двоичных наборов, интерпретируемых как вершины бинарного дерева, включающее пустой набор λ, соответствующий корню этого дерева. Всякая вершина α дерева имеет левого и правого потомков, обозначаемых как $\alpha 0$ и $\alpha 1$. Отдельное сложное знание представляется конечным нагруженным бинарным деревом S. Значение разметки вершины α этого дерева обозначается как $[S]_\alpha$. Множество всех (висячих) вершин дерева S обозначается как $D(S)$ ($O(S)$).

Описание правила составляют четыре следующих раздела: структур образцов (*Ts*), свойств (*Ps*), алгоритмов (*As*), а также синтеза заключения правила (*Cs*). Описания структур образцов составляют именованные образцы $S_1, ..., S_k$, $k \geqslant 0$, для которых дополнительно уточняются именованные классы вершин $Q_{i,1}, ..., Q_{i,n_i}$ и отдельные именованные вершины.

Описание произвольного класса вершин в S_i имеет вид <*имя*>[*описание множества вершин*]. Здесь необязательным параметром *описание множества вершин* реализуется один из способов задания именованного множества вершин. Описание вершины имеет вид <*имя*>[*описание значения*]. Здесь параметр *описание значения* также не обязательный. Он задается выражением, определяющим двоичный набор, представляющий вершину. Раздел свойств образуют логические выражения, выполняющиеся для системы множеств вершин, отдельных вершин и разметок вершин образцов. Они составлены из операций (связок и кванторов), сравнений, имен (образцов, фрагментов образцов и отдельных вершин), конкретных значений, задающих вершины и разметки вершин. Для применения правила к конкретному семейству

фрагментов сначала задается соответствие фрагментов образцам-посылкам. Затем реализуются вызовы всех процедур и проверка выполнимости посылок-свойств рассматриваемого правила. Если свойство содержит параметр, то уточняется множество возможных значений этого параметра, для которых свойство истинно. Раздел алгоритмов правила составляют обращения к конкретным процедурам, задаваемые с помощью конструкции

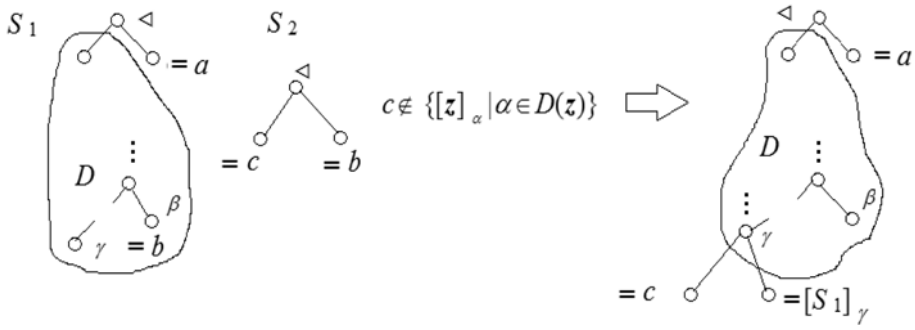


Рис. 6. Правило добавления элемента в синтезируемый фрагмент

<*имя*>:= *Procesure* <*имя процедуры*>
(<*имя_*1>=<*значение_*1>, ..., <*имя_k*>=<*значение_k*>).

Здесь <*имя*> (<*имя_i*>) — это имя некоторой сущности предметной области, значение которой отыскивается с помощью заданной процедуры (сущности, соответствующей элементу семейства параметров процедуры). Обращение к процедуре возможно, если фрагменты конфигураций содержат значения всех параметров для соответствующей процедуры. Если правило содержит обращение к процедуре, то реализация обращения завершается построением фрагмента конфигурации, связывающего понятие, представленное параметром <*имя*>, с вычисленным значением.

Раздел синтеза правила содержит схему конструирования фрагмента знания. Она имеет вид композиции областей и вершин фрагментов знаний, соответствующих областям и вершинам образцов посылок правила.

Рассмотрим пример формализованного описания правила синтеза фрагмента конфигурации для решения задачи $X \lhd a$, связанного с построением иерархической структуры, составленной всеми классами объектов, частями которых являются объекты, относящиеся к понятию *a*. Графическое изображение данного правила приведено на рис. 6. Этим правилом моделируется шаг расширения синтезируемого фрагмента конфигурации, в который добавляется новая вершина. При этом классы вершин в образцах изображаются именованными областями, ограниченными непрерывными замкнутыми линиями, а отдельные вершины — кружками. Вершины могут именоваться. Приписанное произвольной вершине выражение вида =*a* означает, что разметкой изображенной вершины является *a*. Левым и правым потомками внутренней вершины α являются вершины $\alpha 0$ и $\alpha 1$. Область синтеза правила изображается в правой части правила и отделена от остальной части правила символом стрелки.

Изображение правила представляет наглядно структуры фрагментов конфигураций, но является слабо формализованным [3].

Рассматриваемое правило составляют образцы S_1 и S_2, а также сравнение $c \notin \{[z]_\alpha \mid \alpha \in D(z)\}$. Вершина α — самая правая висячая вершина всякого фрагмента конфигурации для образца S_1. Область D составляют вершины поддерева S_1, корень которого является предком висячей вершины β. Образец S_2 соответствует простому знанию. Положение вершины β, удовлетворяющей условиям правила, не является детерминированным. Уточнить схему выбора подходящих вершин для реализации оператора вывода можно, выбрав метод прохождения вершин из рассматриваемого образца. К ним относятся модификации обходов в глубину и ширину (обозначаются как *Depths* и *Width*). Этим обеспечивается детерминированность процесса синтеза фрагментов по образцам заключений.

С каждым образцом связано значение дополнительного атрибута завершенности образца, принимающего значения "*отдельный*" или "*полный*". Для первого значения атрибута образцу соответствует всякий фрагмент конфигурации, имеющий заданную структуру. Во втором случае образец правила помечен справа символом *. Образцу с таким атрибутом соответствует фрагмент конфигурации, который не может быть расширен с сохранением свойств, представленных в правиле. Например, выражение $(S_1)^*$ определяет фрагменты конфигурации, содержащей все возможные элементы, которые могут быть добавлены в объект правой части рассматриваемого правила.

Приведем описание правила, изображенного на рис. 6, с использованием элементов специального языка. Система посылок правила представляется выражением, близким формату структурного представления конфигураций абстрактных пространств знаний:

$$Ts\, S_1$$
$$Classes\left(D = \{x \mid x = \omega\sigma\}\right)$$
$$Vertices(\beta, \gamma)$$
$$Ts\, S_2$$
$$Classes\left(D(S_2) = \{\lambda, 0, 1\}\right)$$
$$Pr\left(\beta \in O(S_1)\right),\ \left([S_1]_\omega = :\right),\ \left(\omega 1 = \beta\right),\ \left(\gamma = \omega(0)^n\right),\ \left(\gamma \in O(S_1)\right),$$
$$\&\left([S_2]_0 \notin D(S_1) \cap I_\omega\right).$$

Включение в описания образцов специальных функций позволяет унифицировать методы задания структурных элементов образцов. Для рассмотренного примера правила такими функциями являются вычисление предшествующей вершины *Up*(α), крайнего слева потомка вершины *Left*(α) и всех потомков заданной вершины дерева *Subset*(α). Тогда система посылок рассматриваемого правила может быть представлена следующим описанием:

$$Ts\ S_1$$
$$Classes\left(D = Subset\left(Up\left(\alpha\right)\right)\right)$$
$$Vertices\left(\beta, \gamma = Left\left(Up\left(\beta\right)\right)\right)$$
$$Ts\ S_2$$
$$Sets\left(D\left(S_2\right) = \{\lambda, 0, 1\}\right)$$
$$Pr\left(\left[S_1\right]_{Up(\beta)} = \vdots, \left[S_1\right]_{\beta} = \left[S_2\right]_1,\ \left[S_2\right]_0 \notin D\right).$$

Образец заключения правила составляется из фрагментов знаний, соответствующих элементам образцов посылок, в которых также полезны специальные функции, моделирующие действия, связанные с добавлением и удалением фрагментов. Описание заключения для рассматриваемого примера моделирует замену элемента γ образца S_1 на фрагмент, представляющий композицию $(c \circ \vdots) \circ \left[S_1\right]_{\gamma}$. Схема составления заключения правила использует операцию замены вершины на дерево, задаваемую выражением $Change\left(S_1, \gamma, (c \circ \vdots) \circ \left[S_1\right]_{\gamma}\right)$.

Обработку семейства правил синтеза конфигураций, составляющих решение отдельной задачи, можно реализовать способом, близким прямому выводу. Сначала конструируется последовательность фрагментов конфигураций, составляющих начало процесса решения задачи. Всякое правило применяется к элементам уже построенной последовательности. Ранжирование правил семейства позволяет уточнять очередность их применения. Например, параметр кратности применения отдельных правил, задаваемый как *Singular* (однократное) или *Multiple* (многократное). Для первого случая правило применяется ровно один раз. Для своего следующего выполнения правило должно быть заново отобрано из множества правил, связанных с решением рассматриваемой задачи. Многократное применение связано с повторным применением правила к последовательно синтезируемым фрагментам пока выполняются условия этого правила.

Оператор вывода моделирует процесс решения задач с помощью механизма управления системой последовательно решаемых подзадач. Одна из схем перехода к подзадачам связана с синтезом конфигурации, соответствующей образцу посылке правила. Для этого требуется решить дополнительную каноническую задачу. Рассмотренный ранее пример правила содержит два образца посылки, проверка которых вызывает переход к подзадачам. Первая связана с построением фрагмента, соответствующего образцу S_1. Вторая задача связана с просмотром простых знаний, представленных образцом S_2. Управление процессом решения задач для нахождения фрагментов знаний, соответствующих образцам, можно реализовать, используя вариант схемы обратного вывода.

Схемы декомпозиции задач на подзадачи

Фрагменты конфигураций, синтезируемые с помощью правил решения задач, могут составляться в несколько этапов, каждый из которых относится к решению подзадачи начальной задачи. Сценарий решения задачи для такого случая связан с декомпозицией на подзадачи. Для управления процессом решения задачи применяется структура декомпозиции, представленная понятиями композиции (◦), варианта (,) и серии (*). В сценарии эта структура реализуется деревом задач вместе с правилами перехода между задачами и подзадачами. Результатом выполнения всякого сценария является синтезированный фрагмент знания. Описание сценария определяет последовательность решаемых подзадач. При этом комбинация $(T_1 \circ T_2)$ означает две последовательно решаемые задачи, так что вторая задача применяется к фрагменту, синтезированному при решении первой задачи. Конструкция (T_1, T_2) означает, что решается одна из двух задач, выбираемая проверкой дополнительного условия на синтезированный фрагмент конфигурации. Серия одинаковых задач $(T)^*$ связана с последовательным повторным решением одной и той же задачи, каждая из которых продолжает решение предыдущей задачи. Например, выражение $(X \lhd a)^*$ соответствует обратной задаче, решением которой является весь фрагмент отношения $\lhd$, образующий нижний конус элемента *a*, выводимый из модели области знаний.

Рассмотрим пример решения задачи синтеза фрагмента конфигурации, использующий декомпозицию на подзадачи, приведенный на рис. 7. Основная задача относится к классу задач проектирования и состоит в построении модели объекта, соответствующего системе инженерных знаний для понятия "*жилой дом*" [5]. Начальная постановка имеет вид $T(дом \equiv X)$. Решения этой задачи связаны с нахождением значения фрагмента, обозначенного как *X*. Декомпозицию данной задачи составляют подзадачи построения полного фрагмента отношения "*быть частью*" для выбранного понятия. Данная задача представляется выражением $T_1\left((дом \lhd Y)^*\right)$, а ее решение устанавливает морфологическую структуру синтезируемого объекта, которое подставляется в исходную задачу в качестве *X*. Следующая решаемая задача связана с интеграцией в создаваемую структуру систем атрибутов для всех понятий, представленных висячими вершинами построенного фрагмента отношения агрегирования [3]. Синтез систем

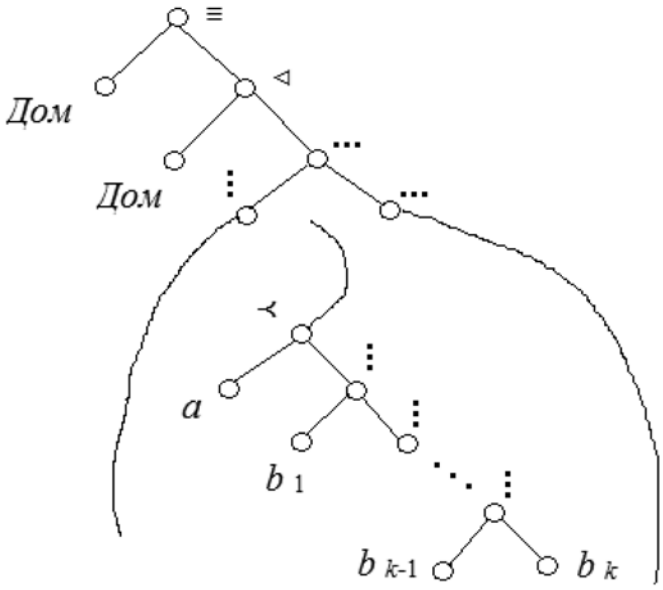


Рис. 7. Фрагмент конфигурации, синтезируемой для агрегирования подзадач T_1 и T_2

атрибутов осуществляется через решение серии подзадач, предписанной выражением

$$T_2\Big(\big([z_1]_a \prec X\big)^*;\ a \in O(z);$$
$$Depth;\ Up - Dn;\ L - R\Big).$$

где α — произвольная висячая вершина — понятие из структуры решения первой задачи, и такие вершины перебираются в порядке обхода в глубину (*Depth*) сверху вниз (*Up—Dn*)) и слева направо (*L—R*). Решения рассматриваемой серии подзадач встраиваются в синтезируемый фрагмент конфигурации.

После получения полного решения второй подзадачи реализуется следующая задача, связанная с нахождением и согласованием значений атрибутов структурных частей. Общий формат такой задачи, унифицированный ко всем атрибутам, имеет вид

$$T_3\Big(\big([z_1]_a = X\big)^*;\ a \in O(z_2)\setminus O(z_1);$$
$$Depth;\ Up - Dn;\ L - R\Big).$$

Извлечение решений из синтезируемых конфигураций

Трассирование является формальной конструкцией, уточняющей схему извлечения решения заданной задачи из синтезированного фрагмента конфигурации [2]. Многообразие трассирований иерархических семантических структур разбивается на классы, из которых специальный интерес представляют трассирования расширения и сжатия, а также класс общих трассирований, допускающих оба названных вида трассирований. Правилами синтеза фрагментов знаний для задач разных типов обеспечивается распознавание возможности извлечения решений задач из таких фрагментов с помощью подходящих видов трассирований.

Заключение

Создание формальной теории является единственным способом точной формализации понятий, относящихся к выбранной области деятельности [2]. Это в полной мере относится к области искусственного интеллекта, развивающейся в значительной степени индуктивно через многообразие эмпирических порождающих принципов, составляющих слабо формализованные кибернетические модели, уточняемые для конкретных прикладных интеллектуальных систем [6]. Приобретаемый при этом опыт систематизируется в форме разнородных эмпирических знаний о реализованных подходах, уточнения которых не претендуют на унифицированность или универсальность. Построение универсальных инвариантов для структурных и функциональных аспектов интеллектуальных систем позволяет предложить математическую теорию, класса систем, представлениями которых являются как известные, так и возможные подходы к построению систем, основанных на знаниях. Это позволяет рассматривать разные формализмы как реализации одних и тех же порождающих принципов, способствуя эффективному сравнению и совместному использованию конкретных формализмов знаний.

Работа выполнена при поддержке РФФИ грант № 16-01-00214.

Список литературы

1. **Чёрч А.** Введение в математическую логику. Т. 1. М.: И. Л. 1960. 478 с.

2. **Костенко К. И.** Формализмы представления знаний и модели интеллектуальных систем. Краснодар: Кубанский гос. ун-т, 2015. 300 с.

3. **Костенко К. И.** Правила оператора вывода абстрактного пространства знаний // Программная инженерия. 2016. Т. 7, № 6. С. 258—267.

4. **Джексон П.** Введение в экспертные системы. М.: Вильямс, 2001. 624 с.

5. **Янов Ю. И.** Математика и метаматематика // Математические вопросы кибернетики. 2007. Вып. 16. С. 129—154.

6. **Чечкин А. В.** Активирующая подсистема интеллектуального управления и теория функциональных систем П. К. Анохина, URL: http://www.keldysh.ru/pages/BioCyber/RT/Chechkin.htm

Simulation of Inference Operator for Hierarchical Knowledge Representation Formalisms

K. I. Kostenko, kostenko@kubsu.ru, Kuban State University, Krasnodar, 350040, Russian Federation

Corresponding author:

Kostenko Konstantin I., Head of Chair, Kuban State University, Krasnodar, 350040, Russian Federation, e-mail: kostenko@kubsu.ru

Received on May 31, 2016
Accepted on June 30, 2016

Hierarchical knowledge representation formalisms provide more efficient tool for simulating the knowledge domains contents models compared to descriptive logics based models. These formalisms allow constructing complicated semantic structures, associated with given professional problems that needed searching for the solutions. Application of hierarchical formalisms to professional activity domains is associated with claim of formalization the inference operator concept, as the foundation for realization the decision searching processes. The universal unified inference operator definition is introduced. It uses task structure invariants and complex knowledge, synthesized by processing the subject area knowledge base. Such a base is represented by enumerable sets of elementary and simple knowledge extracted from subject domain content. The possibility of any task reducibility into tasks combination from four special classes is settled. Every inference operator allows implementation as set of tasks solving processes described by knowledge synthesis rules. Such an operator is supplied by knowledge structure tracing mappings as the tool for tasks solutions extracting from synthesized knowledge. Universal concepts and invariants are proposed for tasks decomposition and knowledge synthesis rules descriptions. The tasks structures and tasks solving processes rules serve to be the foundation for inference operator formal description. The rule premises types are represented by templates, conditions and procedures calls as well as rules' conclusions templates. Rules' premises templates provide components recognitions and analysis for sets of associated knowledge. Conditions and procedures calls allow describing complex situation of knowledge synthesis as specified at rules' conclusions. Task structure is created by the task decomposition process. It defines the subtasks based processes that implement the scenario of tasks solving.

Keywords: *knowledge area, knowledge space, task structure, elementary knowledge, simple knowledge, knowledge synthesis, inference operator*

Acknowledgements: *This work was supported by the Russian Foundation for Basic Research, project nos. 16-01-00214*

For citation:

Kostenko K. I. Simulation of Inference Operator for Hierarchical Knowledge Representation Formalisms, *Programmnaya Ingeneria,* 2016, vol. 7, no. 9, pp. 424—431.

DOI: 10.17587/prin.7.424-431

References

1. **Church A.** *Vvedenie v matematicheskuju logiku* (Introduction to mathematical logic), vol. 1, Moscow, Inostrannaja literatura, 1960, 478 p. (in Russian).

2. **Kostenko K. I.** *Formalizmy predstavlenija znanij i modeli intellektualnyh sistem* (Knowledge representation formalisms and intelligent systems models), Krasnodar, Kuban state university, 2015, 300 p. (in Russian).

3. **Kostenko K. I.** Pravila operatora vyvoda abstraktnogo prostranctva znanij (The rules for abstract knowledge spaces inference operator), *Programmnaya Ingeneria,* 2016, vol. 7, no. 6, pp. 258—267 (in Russian).

4. **Jackson P.** *Vvedenie v Jekspertnye sistemy* (Introduction to expert systems), Moscow, Williams, 2001, 623 p. (in Russian).

5. **Janov Ju. I.** Matematika i metamatematika (Mathematics and metamathematics), *Mathematical problems of cybernetics*, 2007, vol. 16, pp. 129—154 (in Russian).

6. **Chechkin A. V.** *Aktivirujushhaja podsistema intellectual'nogo upravenija i teorija funkcional'nyh sistem P. K. Anohina* (Activating subsystem of intellectual management and P. K. Anohin theory of functional systems), available at: http://www.keldysh.ru/pages/BioCyber/RT/Chechkin.htm (in Russian).

Информационно-аналитическое агентство
TelecomDaily
проводит Международную конференцию

"Развитие OTT услуг в России — OTT Russia 2016",

которая состоится 19 октября 2016 г. в отеле "Holiday Inn Suschevsky" по адресу: г. Москва, ул. Сущевский Вал, д. 74.

Мало кто из специалистов инфокоммуникационной отрасли не слышал об OTT. Эта тема в последнее время волнует не только федеральных и региональных операторов и вещателей, но и контент- и сервис-провайдеров. Большое внимание уделяется международному опыту, где OTT-услуги начали свое наступление несколько лет назад и в большинстве случаев оказали большое влияние на телекоммуникационный и вещательный рынки, что в свою очередь вызвало необходимость разработки принципиально новой стратегии для операторов мобильной и фиксированной связи, кабельного ТВ и СНТВ, а также вещателей. Операторы "большой четверки" объединились в альянс чтобы стандартизировать и сделать совместимой работу IP-интерфейсов интернет-сервисов.

Конференция станет экспертной платформой для обсуждения всех необходимых изменений и дополнений в действующее законодательство, вопросов стандартизации и совместимости услуг и пользовательских устройств, что послужит важнейшим фактором развития Электронных коммуникаций, формирования информационного общества и перехода к развитию цифровой экономики страны.

На конференции предполагается обсудить следующие вопросы:

- Рынок OTT в России и мире — текущее состояние и перспективы развития
- Цели и задачи Рабочей группы All IP Interconnect Russia
- Вопросы межоператорского взаимодействия
- Стандартизация IP-интерфейсов, единый профиль для абонентских устройств: IP-совместимости и интеграции VAS-сервисов
- Государственное регулирование и правовые аспекты предоставления All IP-сервисов и OTT-услуг
- OTT-сервисы: большие возможности или реальная угроза?
- Формирование OTT услуг для различных сегментов рынка
- Конвергенция услуг и их предоставление на любом устройстве через любую сеть
- Модели партнерства оператора и провайдера в OTT-сегменте
- Применение OTT-технологий для реализации легитимных бизнес-моделей предоставления интерактивных видео-сервисов
- Позиция производителей ИКТ-оборудования и пользовательских устройств
- Влияние OTT-рынка на производство и дистрибуцию контента в России

В конференции примут участие представители государственных регулирующих органов, российских и зарубежных операторов связи, вещателей, кабельных ТВ и СНТВ-сетей, сервис- и контент-провайдеров, разработчиков и поставщиков решений, производителей телекоммуникационного оборудования, системных интеграторов, корпоративных служб связи и инфокоммуникационных услуг, инвестиционных компаний, а также консультанты и эксперты отрасли, отраслевые СМИ, представители компаний потребителей услуг связи.

Сайт конференции: http://www.tmtconferences.ru/ott2016.html
Все предложения и пожелания направляйте на адрес электронной почты conf@tdaily.ru

ООО "Издательство "Новые технологии". 107076, Москва, Стромынский пер., 4
Технический редактор *Е. М. Патрушева.* Корректор *Т. В. Пчелкина*

Сдано в набор 07.07.2016 г. Подписано в печать 19.08.2016 г. Формат 60×88 1/8. Заказ PI916
Цена свободная.

Оригинал-макет ООО "Адвансед солюшнз". Отпечатано в ООО "Адвансед солюшнз".
119071, г. Москва, Ленинский пр-т, д. 19, стр. 1. Сайт: **www.aov.ru**