

**Шалыева Ирина Михайловна** – e-mail: ishalyaeva@bk.ru; кафедра информационных технологий в бизнесе; студентка.

**Aizatullova Renata Rinatovna** – National Research University “Higher School of Economics”; e-mail: ajz-renata@mail.ru; 37a, Gagarina St., Perm, 614060, Russia; phone: +73422825372; the department of business information technologies; student.

**Lyadova Lyudmila Nickolaevna** – e-mail: LLyadova@hse.ru; the department of business information technologies; candidate of sciences; associate professor.

**Sukhov Alexander Olegovich** – e-mail: sukhov\_psu@mail.ru; the department of business information technologies; cand. of phis.-math. sc.; senior teacher.

**Shaliaeva Irina Mihailovna** – e-mail: ishalyaeva@bk.ru; the department of business information technologies; student.

УДК 004.4'22

**Ю.И. Рогозов, А.А. Дегтярев**

### **РАЗРАБОТКА ОСНОВ АЛГЕБРЫ НАД ОПЕРАЦИОННЫМИ ПРОГРАММАМИ\***

*Рассматривается проблема формализации процесса создания программ потоковой обработки данных. Исследуются существующие способы представления механизмов реализации действий. Рассматривается природа программ потоковой обработки данных и их основные составляющие – операционные программы. Вводится понятие механизма создания программ потоковой обработки данных, который должен содержать в себе в качестве элементов операционные программы, а в качестве функций, операции позволяющие соединять элементы различным способом. Рассматриваются основные классы операционных программ и характерные для них особенности внутренней и внешней организации. Поскольку классическое понятие множества, по своему определению, не накладывает никаких ограничений на природу предметов входящих в него, оно не может использоваться для формализации рассматриваемого механизма. Поэтому предлагается математический формализм – алгебра над операционными программами, которая представлена совокупностью носителя алгебры – многослойное множество, и сигнатурой алгебры – операции определенные на носителе. Данный формализм ввиду своей специфики позволяет учитывать внутреннюю организацию операционных программ, что предоставляет возможности для проработки вопросов автоматизированного создания программ потоковой обработки данных.*

*Конфигурирование; механизм реализации действия; программы потоковой обработки данных; алгебра над программами; формализация.*

**Y.I. Rogozov, A.A. Degtyarev**

### **THE DEVELOPMENT FOUNDATION OF ALGEBRA ON OPERATIONAL PROGRAMMS**

*In this paper is consider problem of formalizing the process of creating programs streaming data. We investigate ways of presenting existing implementation mechanisms of action. Examines the nature of streaming data processing programs and their main components – the operating program. Introduces the concept of a mechanism for creating streaming data processing programs, which should contain as elements of operational programs, and as functions, allowing the operation to combine the elements in a different way. The basic classes of operational programs and their particular characteristics of internal and external organizations is consider. Since the*

\* Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 13-07-00971.

*classical notion of a set, by definition, does not impose any restrictions on the nature of the items included in it, it can not be used to formalize the mechanism under consideration. It is therefore proposed mathematical formalism – algebra on operational programs, which is represented by a set of carrier algebra – multi-set, and the signature of the algebra – certain operations on the carrier algebra. This formalism because of its specificity allows to consider the internal organization of operational programs that provide opportunities for the study of questions automated creation programs streaming data.*

*Configuration; implementation mechanism of action; stream-processing program; algebra over programs; formalization.*

**Введение и постановка задачи.** В настоящее время в практических задачах программной инженерии наблюдаются тенденции к построению предметно-ориентированных программных фреймворков (каркасов), которые позволяют существенно снизить стоимость создания информационных систем определяемого фреймворком класса (семейства систем) [1, 2, 3, 4]. В то же время в работе [5, 6] предметно ориентированные фреймворки рассматриваются как метасистемы, которые по своей сути являются системами с динамической структурой или с концептуальной точки зрения деятельностью с динамической структурой. При таком подходе система, решающая конкретную задачу и являющаяся порождением метасистемы, может быть концептуально определена как деятельность со статически заданной структурой. Процесс такого перехода от динамики к статике в практике обычно называют конфигурированием.

Для того чтобы описать/зафиксировать деятельность с динамической структурой в работах [5, 6], предлагается использовать понятие механизма реализации действия, поскольку любая деятельность может быть рассмотрена как действие для деятельности более высокого уровня. Механизм представляемый в виде набора списков экземпляров характеристик действия позволяет статически изобразить возможную вариативность в действии. Простейшим примером может быть механизм использующий два типа характеристик действия: элементы и функции над элементами. Стоит заметить, что такое определение механизма соответствует математическому понятию алгебры: «Алгеброй  $A$  называется совокупность  $\langle \rangle$  множества  $M$  с заданными в нем операциями  $S \dots A = \langle M, S \rangle$ » [7].

В данной статье рассматривается механизм, результатами которого являются законченные программы потоковой обработки данных, примерами таких программ могут быть тракты обработки или управления в гидроакустических информационных системах [8, 9, 10].

Обычно такие законченные программы потоковой обработки данных строятся из более мелких и универсальных подпрограмм, которые часто называют операционными программами. Операционные программы в общем случае представляют собой элементы преобразующие вектор входных переменных в вектор выходных переменных, по определенным правилам и алгоритмам. Очевидно, что если рассматривать ограниченную предметную область (например, обработка данных в гидроакустических системах), то для решения практически любых задач связанных с обработкой можно сформировать ограниченный и достаточный набор операционных программ. Также ясно, что формализация механизма построения законченных программ потоковой обработки позволит заложить формальный базис, необходимый для автоматизации процесса их построения. Поскольку классическое понятие множества, по своему определению, не накладывает никаких ограничений на природу предметов входящих в него, оно не может использоваться для формализации рассматриваемого механизма. Поэтому необходимо разработать такой формальный базис, который бы позволял при формализации механизмов различных действий учитывать содержание предметов рассматриваемых как элементы механизма.

Данная статья посвящена вопросам создания такого формального базиса.

**Разработка основ алгебры над операционными программами.** В качестве элементов рассматриваемого механизма создания программ потоковой обработки данных выступают операционные программы, выполняющие различные функции. По этой функциональной принадлежности все элементы разбиваются на ограниченное число категорий (данные, операции над данными, управляющие программы), которые в свою очередь могут быть разделены на различные классы (источники скалярных данных; источники векторных данных; преобразователи данных; приемники векторных данных; векторные буферы; скалярные буферы). Каждый класс имеет свои особенности, связанные с организацией входных, выходных и внутренних данных, например элемент относящийся к классу источников векторных данных, может содержать только выход, определяемый как минимум двумя переменными – указатель на массив и размер данного массива. В свою очередь преобразователи данных должны иметь как входы, так и выходы, на которые не накладываются ограничения. Очевидно, что в качестве функций рассматриваемого механизма, необходимо рассматривать операции позволяющие организовывать различные цепочки преобразований из имеющихся операционных программ. Таких операций должно быть две: первая позволяет осуществить последовательное соединение двух элементов по их «выходу-входу»; вторая должна осуществлять объединение выходов двух различных элементов для возможности реализации параллельных цепочек вычислений. Прежде чем вводить правила выполнения функций над элементами, необходимо выбрать математический способ представления элементов, т.е. задать носитель алгебры.

Исходя из изложенного представления об основных классах элементов механизма нижнего уровня, все элементы предлагается отобразить в некоторые абстрактные математические модели.

Пусть задано некоторое множество элементов  $\xi = \{x_i\}$ , и символ  $\theta$  обозначающий признак неиспользуемости. Пусть на множестве  $\xi \cup \theta$  задано множество всевозможных отношений эквивалентности  $R = \{R_k\}$ . Каждое такое отношение эквивалентности разбивает  $\xi \cup \theta$  на семейство непустых множеств  $\{X_\alpha^k\}_{\alpha \in A}$  таких, что

$$\begin{aligned} X_\alpha^k \cap X_\beta^k &= \emptyset; \forall \alpha, \beta \in A; \alpha \neq \beta; \\ \xi \cup \theta &= \bigcup_{\alpha \in A} X_\alpha^k, \end{aligned} \quad (1)$$

где  $A$  – некоторое множество индексов для обозначения блоков разбиения  $k$ .

Определим еще одно семейство непустых множеств  $E^k \subset \{X_\alpha^k\}_{\alpha \in A}$ , в которое входят все  $X_\alpha^k$  не содержащие символа  $\theta$ , т.е.

$$(\exists X_\alpha^k, X_\alpha^k \cap \theta \neq \emptyset) \leftrightarrow (X_\alpha^k \notin E^k). \quad (2)$$

Пусть существует функция  $F$  назначающая блокам подмножества  $E^k$  разбиения  $\{X_\alpha^k\}_{\alpha \in A}$  один из двух знаков {"Входной", "Выходной"}.

**Определение 1.** Многослойным множеством  $P$  называется множество таких объектов  $\eta$  состоящих из разбиения  $E^k$  и функции  $F$ , т.е.

$$P = \{\eta_\gamma\}, \eta_\gamma = \langle E^k, F \rangle. \quad (3)$$

Поскольку  $E^k$  – это семейство множеств являющееся разбиением по определению, то составляющие его члены будем называть блоками. Для обозначения блока принадлежащего  $E^k$  будем использовать следующую запись  $X_{l,j}^k \in E^k$ , где  $l$  – это мощность рассматриваемого блока, а  $j$  – его уникальный идентификатор среди всех блоков всех разбиений, т.е.  $j = 1, \dots, J$ , где

$$J = |\bigcup_\gamma E^k \in \eta_\gamma|. \quad (4)$$

**Примечание.** Предполагается, что блоки, принадлежащие разным разбиениям, но составленные из одного и того же набора элементов, считаются совпадающими или подобными. При этом нижние индексы двух блоков совпадают, т.е.

$$\forall x_i [x_i \in X_{l,j}^k \leftrightarrow x_i \in X_{l,p}^m] \leftrightarrow (X_{l,j}^k = X_{l,p}^m)(l = p), k \neq m. \quad (5)$$

Если блоку  $X_{l,j}^k$  функцией F назначен входной знак, то такой блок будем обозначать диакритическим символом  $\overrightarrow{X_{l,j}^k}$ . Аналогично будет обозначаться блок, которому назначен выходной знак  $\overleftarrow{X_{l,j}^k}$ .

Таким образом, элемент многослойного множества (ЭММ) можно представлять как набор блоков из разбиения  $\Xi^k$  с сигнатурой определяемой функцией F, т.е.

$$\eta = \{\overrightarrow{X_{l,j}^k}\}, \quad (6)$$

где  $X_{l,j}^k \in \Xi^k$ , а диакритический символ  $\overrightarrow{\quad}$  и  $\overleftarrow{\quad}$  может принимать два различных значения в зависимости от знака блока  $\overrightarrow{\quad}$  и  $\overleftarrow{\quad}$ , входные и выходные знаки соответственно. В дальнейшем будем придерживаться приведенного обозначения элемента многослойного множества.

**Определение 2.** Два блока  $\overrightarrow{X_{l,j}^k}$  и  $\overrightarrow{X_{t,p}^m}$ , принадлежащие разным ЭММ, называются *подобными*, если совпадают их нижние индексы, т.е.  $l = t$  и  $j = p$ .

**Определение 3.** Два блока  $\overrightarrow{X_{l,j}^k}$  и  $\overrightarrow{X_{t,p}^m}$ , принадлежащие разным ЭММ, называются *парными*, если они являются *подобными* и имеют одинаковую *сигнатуру*.

**Определение 4.** Два блока  $\overrightarrow{X_{l,j}^k}$  и  $\overrightarrow{X_{t,p}^m}$ , принадлежащие разным ЭММ, которые не являются *подобными*, считаются *различными*.

ЭММ удобно использовать в качестве спецификации для любых элементов механизма создания программ потоковой обработки данных. В этом случае различные информационные потоки элементов механизма определяются в виде блоков, которые в свою очередь однозначно определяются посредством набора элементов  $x_i$ .

Информационные потоки элементов механизма также специфицируются по своей направленности – входные и выходные, что соответствует знакам, которые назначены блокам. В выражении 7 приведен пример произвольного ЭММ, в котором раскрыто содержание блоков:

$$\eta = \left\{ \underbrace{\{x_1, x_2\}}_{\overleftarrow{X_{2,1}}}, \underbrace{\{x_3, x_4, x_5\}}_{\overleftarrow{X_{3,1}}}, \underbrace{\{x_6, x_7, x_8\}}_{\overleftarrow{X_{3,2}}} \right\}. \quad (7)$$

Введем определения необходимых операций над ЭММ: операция объединения и операция соединения.

**Определение 5.** Операцией объединения  $\eta_1 \cup \eta_2$  двух ЭММ  $\eta_1$  и  $\eta_2$  называется такая операция, в результате которой формируется новый ЭММ  $\eta_3$ , который состоит из всех *различных* и *подобных* блоков  $\eta_1$  и  $\eta_2$ , *парные* блоки добавляются в новый ЭММ в одном экземпляре. По своей сути данная операция аналогична классической операции объединения в теории множеств.

**Определение 6.** Операцией соединения  $\eta_1 \oplus \eta_2$  двух ЭММ  $\eta_1$  и  $\eta_2$  называется такая операция, в результате которой формируется новый ЭММ  $\eta_3$ , который состоит из всех *различных* блоков  $\eta_1$  и  $\eta_2$ , *парные* блоки добавляются в новый ЭММ в одном экземпляре, а *подобные* блоки поглощаются друг другом и не добавляются в новый ЭММ. Для данной операции существует два ограничения:

**Ограничение 1.** Существует такая пара элементов ЭММ, у которых все составляющие их блоки являются *подобными*, применение операции соединения к такой паре элементов в результате вместо нового ЭММ даст пустое множество.

**Ограничение 2.** Теоретически любой ЭММ может иметь одну из следующих трех сигнатур: (-1) – выход одного разбиения; (0) – выход одного и вход одного или нескольких разбиений; (1) – вход одного или нескольких разбиений. Тогда для операции соединения двух ЭММ действуют правила, определяемые матрицей разрешенных связей (табл. 1), где с помощью н/д обозначены недопустимые сочетания ЭММ для соединения.

Таблица 1

**Матрица разрешенных связей между различными ЭММ**

	(1)	(-1)	(0)
(1)	(1)	(0)	(1)
(-1)	(0)	н/д	(0)
(0)	(1)	(0)	н/д

**Заключение и дальнейшие исследования.** Предложенные основы алгебры над операционными программами позволяют формализовать механизм создания программ потоковой обработки данных из некоторого ограниченного количества готовых операционных программ. Введенная формализация позволяет отображать любую операционную программу в виде семейства множеств. При этом каждое подмножество из семейства множеств характеризует состав информационного потока и называется **блоком**. Один и тот же блок может быть помечен как входной или выходной, что позволяет отображать направленность информационного потока. Две введенные операции «объединение» и «соединение» обеспечивают организацию как последовательных, так и параллельных цепочек операционных программ, а также позволяют осуществлять контроль за передачей информационных данных и параметров. В будущем планируется более детально исследовать возможные ограничения на сочетания различных ЭММ для операций «объединение» и «пересечение» и разработать программную реализацию предлагаемой алгебры над операционными программами.

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Чернецки К., Айзенкер У. Порождающее программирование: методы, инструменты, применение: Пер. с англ. – СПб.: Питер, 2005. – 736 с.
2. Rogozov Ю.И., Свиридов А.С., Дегтярев А.А. Метрики для оценки потенциала к формированию линейки продуктов программного обеспечения из набора похожих программных продуктов // Технологии разработки информационных систем: Сборник материалов Международной научно-технической конференции. Т. 1. – Таганрог: Изд-во ГТИ ЮФУ, 2011. – С. 78-82.
3. Matthias Kowal, Ina Schaefer, Mirco Tribastone. Family-Based Performance Analysis of Variant-Rich Software Systems // Fundamental Approaches to Software Engineering Lecture Notes in Computer Science. – 2014. – Vol. 8411. – P. 94-108.
4. Rogozov Ю.И., Свиридов А.С., Дегтярев А.А. Построение классификации характеристик программного обеспечения с целью идентификации понятий предметной области как характеристик // Информатизация и связь. – 2011. – № 3. – С. 80-83.
5. Rogozov Ю.И. Подход к определению метасистемы как системы // Труды ИСА РАН. – 2013. – Т. 63, № 6. – С. 92-110.
6. Rogozov Ю.И. Понятие метасистемы как методологической основы создания системы // Промышленные АСУ и контроллеры. – 2013. – № 2. – С. 38-46.
7. Горбатов В.А. Основы дискретной математики. – М.: Высш. школа, 1986. – 311 с.
8. Дегтярев А.А. Архитектура программного ядра для построения информационных систем сбора, обработки и хранения данных на примере малоканальных гидроакустических систем // Высокопроизводительные вычислительные системы: Сб. науч. тр. Вып. 2. – Таганрог: Изд-во ЮФУ, 2012. – С. 711.

9. *Дегтярев А.А., Дегтярева Е.Е., Rogozov Ю.И.* Конфигурируемый каркас для гидроакустических компьютеризированных приборов // Свидетельство о государственной регистрации программы для ЭВМ. №201366087 от 23 октября 2013 года.
10. *Дегтярев А.А.* Разработка инвариантного ядра программного обеспечения гидроакустических информационных систем // VIII Ежегодная научная конференция студентов и аспирантов базовых кафедр Южного научного центра РАН: Тезисы докладов (11-26 апреля 2012 г., г. Ростов-на-Дону). – Ростов-на-Дону: Изд-во ЮНЦ РАН, 2012. – С. 123-125.

Статью рекомендовал к опубликованию д.т.н., профессор В.Н. Иванченко.

**Рогозов Юрий Иванович** – Южный федеральный университет; e-mail: rogozov@tti.sfedu.ru; 347928, г. Таганрог, ГСП 17-А, Некрасовский, 44; тел.: 88634371787; кафедра системного анализа и телекоммуникаций; зав. кафедрой; д.т.н.; профессор.

**Дегтярев Алексей Андреевич** – e-mail: alexey.a.degtyarev@gmail.com; кафедра системного анализа и телекоммуникаций; аспирант.

**Rogozov Yury Ivanovich** – Southern Federal University; e-mail: rogozov@tti.sfedu.ru; 44, Nekrasovsky, GSP 17-A, Taganrog, 347928, Russia; phone: +78634371787; the department of system analysis and telecommunications; head the department; dr. of eng. sc.; professor.

**Degtyarev Alexey Andreevich** – e-mail: alexey.a.degtyarev@gmail.com; the department of system analysis and telecommunications; postgraduate student.