

## ПРИМЕНЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ МОДЕЛЕЙ ДЛЯ СОЗДАНИЯ РЕДАКТОРА ДИАГРАММ

П. А. СТЕПАНОВ, М. Ю. ОХТИЛЕВ

*Санкт-Петербургский государственный университет аэрокосмического приборостроения,  
190000, Санкт-Петербург, Россия  
E-mail: pavel@stepanoff.info*

Рассматриваются модели и методы описания формального синтаксиса визуального языка. Используется адаптированная вычислительная модель Тыгу, на основе которой созданы универсальный редактор и формальное описание ER-диаграмм для него.

**Ключевые слова:** диаграмма, визуальный язык, визуальный синтаксис, вычислительная модель Тыгу, универсальный редактор диаграмм

В современных условиях постоянно возрастает объем получаемой человеком информации, поэтому крайне актуальной является задача по упрощению ее восприятия. Один из способов решения этой задачи — представление данных в виде рисунков и схем. В ряде областей этот подход практически вытеснил все остальные. В частности, применительно к разработке программного обеспечения (ПО) следует отметить ER-диаграммы [1] и язык UML [2], являющиеся основными средствами анализа и проектирования.

Применение диаграмм, однако, связано с определенными проблемами. Во-первых, даже часто применяющиеся диаграммы требуют использования дорогого ПО. Во-вторых, для новых диаграмм необходимо создавать специальное инструментальное средство. Для решения этих проблем требуется разработка формального описания визуального языка диаграмм и создание ПО для генерации редактора диаграмм по этому описанию. Существующие универсальные средства поддержки произвольных визуальных языков либо очень сложны, либо имеют существенные ограничения по поддержке синтаксиса.

В настоящей статье продемонстрирован подход на основе моделей, сходных с вычислительными моделями Тыгу [3], и показана его применимость к созданию простого редактора ER-диаграмм.

Существует множество формализмов для создания универсального редактора диаграмм. Основным из них являются графовые грамматики — адаптированные формальные подходы для описания текстовых языков [4]; известны и другие подходы — на основе теории категорий [5],  $\lambda$ -исчисления [6] и пр. При этом можно выделить две общие проблемы. Прежде всего, это отсутствие у диаграмм начала и конца, что создает трудности при выборе точки начала анализа; другим недостатком является характерное деление диаграммы на содержащий графическую составляющую пространственный граф отношений (ПГО) и содержащий смысловую нагрузку абстрактный граф синтаксиса (АГС) с последующим анализом только АГС. Все это не способствует широкому использованию универсальных редакторов.

Естественной альтернативой указанным подходам является определение визуального языка на основе ПГО. Для этого можно составить систему уравнений, связывающих параметры различных графических элементов, которая образует формализованную семантическую сеть, подобную вычислительной модели Тыгу.

Приведем определения диаграммы и визуального языка, принятые в работе [7].

Определение 1. Диаграммой называется пара  $D = \{I, R\}$ , где  $D$  — диаграмма,  $I$  — множество изображений,  $R$  — множество отношений между этими изображениями.

Определение 2. Визуальным языком называется множество всех допустимых для него диаграмм.

Для задания вычислительной модели визуального языка введем следующие определения.

Определение 3. Атрибутом называется пара  $a = \{n, k\}$ , где  $n$  — имя атрибута или  $\emptyset$ , если имя атрибута не задано;  $k$  — значение атрибута или  $\emptyset$ , если значение не определено.

Определение 4. Визуальным примитивом называется пара  $v = \{\hat{A}, f_g(\hat{A})\}$ , где  $\hat{A}$  — множество атрибутов примитива,  $f_g(\hat{A})$  — графическая функция (функция, преобразующая значения атрибутов в изображение).

Определение 5. Диаграммой называется множество  $D = \{A, G, V, R\}$ , где:

$A$  — множество атрибутов, такое что  $A = A_n \cup A_v$ , где  $A_n$  — множество не визуальных атрибутов и  $A_v$  — множество визуальных атрибутов, т.е. существует однозначное отображение из  $A_v$  в  $V$ ;

$G$  — алгебраическая система, заданная на атрибутах;

$V$  — мультимножество визуальных элементов диаграммы,  $v \in V \rightarrow v \in V_T$ , где  $V_T$  — предопределенное множество известных визуальных элементов, для которого верно, что  $\forall a \in \hat{A}, \hat{A} \in v, v \in V \rightarrow a \in A_v, \forall a \in A_v \rightarrow a \in \hat{A} \in v \in V$ , т.е. множество  $A_v$  состоит из тех и только тех атрибутов, которые принадлежат визуальным элементам из  $V$ ;

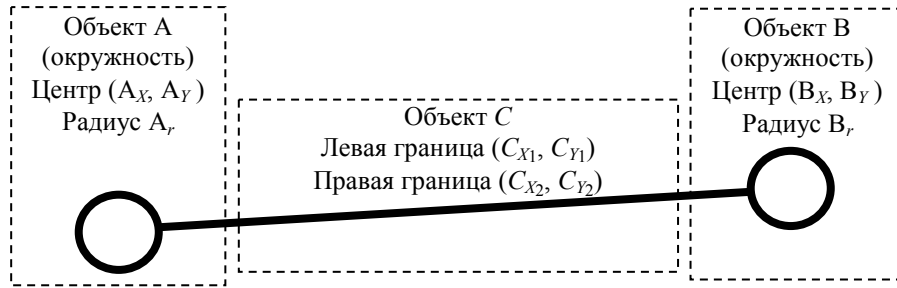
$R$  — множество правил вида  $\text{cond}(c_i): a_i = f_j(A')$ ;  $a_i \in A, A' \subseteq A$ , где  $f_j$  задано в рамках алгебраической системы  $G$ , где  $c_i$  — условие, при истинности которого выполняется присвоение  $a_i = f_j(A)$  (для краткости в случае если условие всегда истинно, правило записывается как  $a_i = f_j(A)$ ).

Как показано в работе [8], вычислительные модели визуального языка образуют строго формализованную семантическую сеть. Пример описания диаграммы такой сетью, состоящей из трех объектов, приведен на рисунке, где  $a$  — визуальные элементы — две окружности и соединяющий их отрезок прямой, а также их представление в виде логических объектов с атрибутами;  $b$  — семантическая сеть, описывающая отношения между атрибутами (прямоугольники, показанные штриховыми линиями) и функциями (прямоугольники из сплошных линий).

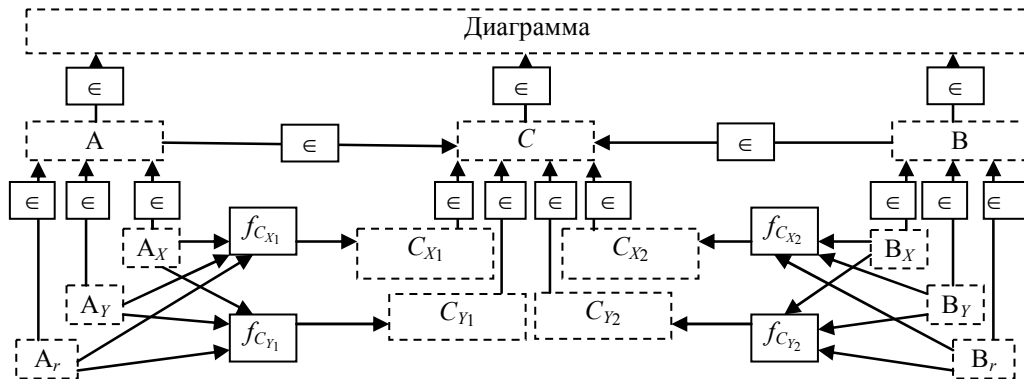
Вычислительные модели Тыгуу эффективны для сравнительно простых задач вида „вычислить значение по набору других значений“, однако в сложной системе с обратными связями модель Тыгуу нуждается в уточнении. Рассмотрим алгоритм вычисления модели, введя предварительно следующее определение.

Определение 6. Воздействием на диаграмму  $D = \{A, G, V, R\}$  называется подмножество множества  $\tilde{A} = \{\tilde{a}_1, \tilde{a}_2, \dots, \tilde{a}_n\}$ , где для любого  $a_i = \{n_i, k_i\} \in A \exists \tilde{a}_i = \{n_i, \tilde{k}_i\} \in \tilde{A}$ , т.е. атрибуты с совпадающими именами имеют, возможно, различные значения.

а)



б)



**Алгоритм** (алгоритм вычисления диаграммы при воздействии). Процесс вычисления состоит из двух этапов. На первом этапе осуществляется первоначальное изменение некоторых переменных  $\tilde{A}$  (воздействие на модель). На втором этапе применяются правила  $R$  до тех пор, пока не будет установлено, что не осталось применимых правил; при этом  $a_i$  добавляется в множество  $\tilde{A}$ . Любое правило  $\text{cond}(c_i): a_i = f_j(A')$ ;  $a_i \in A, A' \subseteq A$  из множества  $R$  может быть применено при соблюдении следующих условий:

- новое значение  $k_i$  еще не получено;
- пересечение  $A'$  и  $\tilde{A}$  не пусто;
- условие  $\text{cond}$  истинно.

Пользователь оперирует объектами. Для этого вводится тип данных — объект диаграммы. Объекты могут владеть атрибутами и другими объектами, в результате чего можно управлять фрагментами диаграммы как единым целым. В рамках модели диаграмма представляется в виде дерева, в корне которого находится диаграмма, а в узлах — принадлежащие ей объекты и атрибуты. Трансляция диаграмм может осуществляться путем обхода этого дерева.

Разработанный редактор состоит из трех основных частей — математического ядра, реализующего вычислительную модель, контроллера представления, выполняющего манипуляции над объектами, и собственно графического интерфейса на Java Swing.

Математическое ядро содержит подсистему вычислений, граф зависимостей и классы, образующие визуальный алфавит. Подсистема вычислений разбирает правые части функций и строит граф зависимостей между атрибутами. При изменении атрибута по графу можно определить перечень необходимых для вычисления функций. Подсистема содержит арифметические и логические функции, скобки и несколько специальных функций; дополнительные функции могут быть дописаны на языке JavaScript. Визуальный алфавит образуется классами, представляющими типовые фигуры (отрезок прямой, окружность, текст и т.п.) и типовые взаимодействия (область перемещения фигуры, область установки связи).

Контроллер представления преобразует описание языка в логику интерфейса. Любое необходимое действие можно представить в виде вычисления математическим ядром

соответствующей функции. Однако некоторые действия являются специфичными для контроллера, например: определение объекта, с которым выполняется взаимодействие, и установление отношений между объектами, которое выполняется на основе областей установки связи и ролей. Область установки связи — это визуальный элемент, содержащий идентификатор, называемый ролью, а также экспортируемые в отношении атрибуты. Объект, вступающий в отношение, предоставляет список ролей других необходимых ему объектов. Контроллер спрашивает пользователя о выборе области установки связи для каждой роли, после чего добавляет объект в модель.

Для проверки рассмотренной модели реализован дескриптор, описывающий простую ER-диаграмму, с возможностью трансляции в код на языке SQL. По результатам сделаны следующие выводы:

- предложенный подход работоспособен для простых визуальных языков;
- логика дескриптора ER-диаграммы перегружена мелкими деталями, поэтому необходимо создание высокоуровневых элементов визуального алфавита;
- полностью отказаться от программирования в пользу математических операций не удалось даже для простой диаграммы;
- типа данных, представляющих список, для работы с наборами данных недостаточно — необходима реализация множеств, мультимножеств и хэш-таблиц.

Исследования, выполненные по данной тематике, проводились при финансовой поддержке ведущих университетов Российской Федерации: Санкт-Петербургского политехнического университета Петра Великого (мероприятие 6.1.1) и Университета ИТМО (субсидия 074-U01), а также Программы НТС Союзного государства „Мониторинг-СГ“ (проект 1.4.1-1), Российского фонда фундаментальных исследований (гранты № 15-07-08391, 15-08-08459, 16-07-00779, 16-08-00510, 16-08-01277), Программы фундаментальных исследований ОНИТ РАН (проект № 2.11), в рамках бюджетных тем № 0073–2014–0009, 0073–2015–0007.

#### СПИСОК ЛИТЕРАТУРЫ

1. *Chen P.* The entity-relationship model-toward a unified view of data // *ACM Transact. on Database Systems.* 1976. Vol. 1, N 1. P. 9—36.
2. *Буч Г., Рамбо Д., Джекобсон А.* Язык UML. Руководство пользователя. М.: ДМК Пресс, 2004.
3. *Тыгу Э. Х.* Концептуальное программирование. М.: Наука, 1984.
4. *Pfaltz J. L., Rosenfeld A.* Web grammars // *Intern. Joint Conf. on AI; Washington, DC, USA.* 1969. P. 600—619.
5. *Ehrig H., Ehrig K., Prange U., Taentzer G.* Fundamentals of Algebraic Graph Transformation. Berlin, Heidelberg, New York: Springer, 2006.
6. *Erwig M.* Functional programming with graphs // *2nd ACM SIGPLAN Intern. Conf. on Functional Programming.* Amsterdam. 1997. P. 52—65.
7. *Chang S-K.* Visual languages: A tutorial and survey // *IEEE Software Magazine.* 1987. Vol. 4, is. 1. P. 29—39.
8. *Степанов П. А., Охтилев М. Ю.* Вычислительная модель визуального языка // *Изв. вузов. Приборостроение.* 2006. Т. 49, № 11. С. 28—32.

#### Сведения об авторах

- Павел Алексеевич Степанов** — СПбГУАП, кафедра компьютерных технологий и программной инженерии; ст. преподаватель; E-mail: pavel@stepanoff.info
- Михаил Юрьевич Охтилев** — д-р техн. наук, профессор; СПбГУАП, кафедра компьютерных технологий и программной инженерии; E-mail: oxt@mail.ru

Рекомендована СПИИРАН

Поступила в редакцию  
01.06.16 г.

**Ссылка для цитирования:** Степанов П. А., Окхтилев М. Ю. Применение вычислительных моделей для создания редактора диаграмм // Изв. вузов. Приборостроение. 2016. Т. 59, № 11. С. 939—943.

**APPLICATION OF COMPUTATIONAL MODELS  
TO CREATION OF UNIVERSAL DIAGRAM EDITOR**

**P. A. Stepanov, M. Yu. Okhtilev**

*St. Petersburg State University of Aerospace Instrumentation,  
190000, St. Petersburg, Russia  
E-mail: pavel@stepanoff.info*

Models and methods of description of visual language formal syntax are considered. An adapted computational model Tyugu is employed; the model is used as a base for development of a universal editor and a formal description of ER-diagram for it.

**Keywords:** diagram, visual language, visual syntax, Tyugu computational model, universal diagram editor

**Data on authors**

- Pavel A. Stepanov** — St. Petersburg State University of Aerospace Instrumentation, Department of Computer Technologies and Software Engineering; Senior Lecturer; E-mail: pavel@stepanoff.info
- Mikhail Yu. Okhtilev** — Dr. Sci., Professor; St. Petersburg State University of Aerospace Instrumentation, Department of Computer Technologies and Software Engineering; E-mail: oxt@mail.ru

**For citation:** Stepanov P. A., Okhtilev M. Yu. Application of computational models to creation of universal diagram editor // Izv. vuzov. Priborostroyeniye. 2016. Vol. 59, N 11. P. 939—943 (in Russian).

DOI: 10.17586/0021-3454-2016-59-11-939-943