

**Единая графическая оболочка для всех языков и систем,
позволяющая программировать всем не только программистам.**

**Вельбицкий И.В.
Фонд Глушкова
Украина
glushkov.org
ivelbit@gmail.com**

Abstract— Визуальная технология программирования Р-схемами нового поколения (ВТР) впервые предлагает не писать, а *рисовать* программы на всем их жизненном цикле, согласно ISO/IEC 8631Н:1989(Е) [3]. Рисовать проще, быстрее, эффективнее, чем это делается сейчас традиционным способом в виде текстов. Это на порядок упрощает и улучшает существующий процесс программирования и позволяет программировать всем не только программистам. В настоящее время реализована графическая среда разработки (не коммерческий вариант), которая имеет 14 визуальных кнопок управления и содержит графический редактор, перевод Р-схем в Р*-схемы (Р-схемы без деталей реализации) и обратно, транслятор РР*-схем в С++ и т.д.

Keywords-Графическое программирование, Р-схема, Р*-схема, Визуальное программирование РР*-схемами, Графическая оболочка, Графическая среда разработки, программирование для всех, нагруженный по лугам граф, параллельное программирование, доказательство правильности программ, метаязык для задания синтаксиса и семантики, сетевой график, автоматическая генерация плана работ.

I. Суть технологии нового поколения

В новой технологии вместо использования традиционных операторов типа **if**, **for**, **goto** и т.д. (**всех!** около 10) предлагается использовать ОДНУ, единственную Р-схему – горизонтальную дугу, которая имеет направление вправо или влево, рис.1. На дуге сверху такого графа ВСЕГДА записывается

Условие прохождения по дуге, а снизу

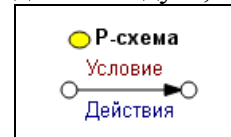


Рис. 1. Р-схема (одна дуга вправо или влево) для записи программы на любом языке.

выполняемые при этом Действия, стрелка дуги указывает следующую вершину для анализа или выполнения. Вершина не имеет имени и задает *состояние* программы или процесса ее разработки. Из вершины может исходить любое число дуг влево и/или вправо. На рис.2 приведена Р-схемы оператора выбора и цикла и соответствующая справа запись в языке С++.

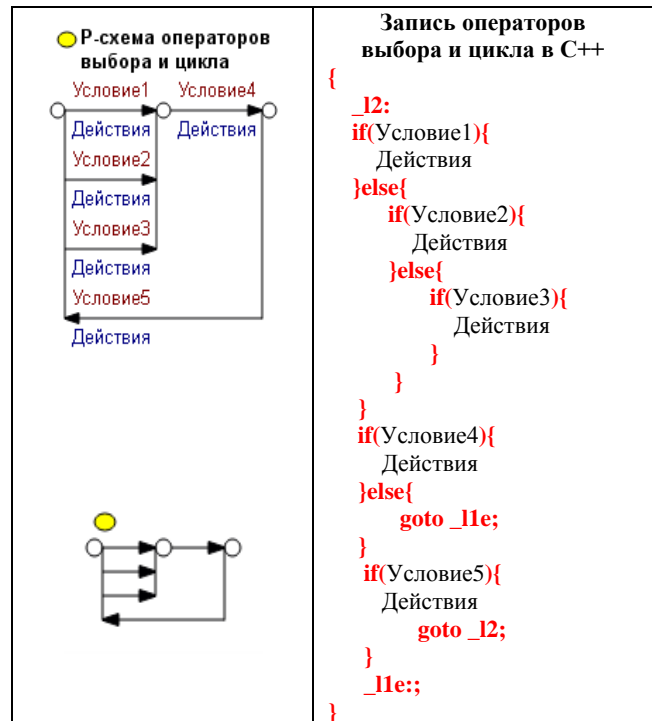


Рис. 2. Запись операторов выбора и цикла в

Р-схемах и в С++. Красным выделены лишние для Р-схемы символы в С++.

Слева внизу на рис.2 приведена Р*-схема, которая является Р-схемой без деталей реализации, без записей на дугах. Это новая форма записи программы, которой нет в традиционном программировании. Она позволяет наглядно и очень компактно (в 50 и более раз для реальных программ) изобразить самую суть, схему, скелет алгоритма. Очень полезная возможность в практике разработки графических программ ВТР.

Р-схема на рис.2 несравненно нагляднее, в два раза компактнее (Р*-схема в 12 раз компактнее) записи программ в С++ и вводится за только 4 (чило горизонтальных дуг) нажатия клавиш мыши или клавиатуры (в 37 раз быстрее ввода С++). Из записи Р-схемы исключена половина лишних символов С++ (на Рис.2 выделены красным). Пользователь ВТР не видит правую часть Рис.2, которая приведена лишь как пояснение Р-схемы в традиционных для читателя обозначениях. Язык записи на дугах может быть любым: русским, английским, китайским, математическим, программистским (computer-sci): Фортран, Кобол, С++, JAVA, Delphi и т.д. (**любой!**) На запись Условий и Действий не накладывается никаких ограничений, она может быть в одну или несколько строк [1-7].

II. РЕАЛЬНЫЙ ПРИМЕР

Такой граф в математике называется графом нагруженным по дугам, в отличие от графа нагруженного по вершинам (всем известные блок-схемы, UML-диаграммы, ДРАКОН-система и т.д. более 100 наименований). Граф в ВТР имеет имя, программа задается ЛЮБЫМ ЧИСЛОМ взаимосвязанных по именам таких графов. На Рис.3 приведена Р*-схема реальной графической ООП-программы, которая осуществляет чтение специального типа файлов в языке **Delphi** и определяется восьмью Р-схемами [8]. На Рис. 3 первой Р-схемой задается архитектура всего ООП-проекта. Первой дугой в этой Р-схеме задается имя класса: **Type=class**. Второй колонкой пять дуг определяют ПОЛЯ класса: **TextFile, array of string, Boolean** и т.д. При задании (описании) данных в ПОЛЕ класса сверху на дуге в

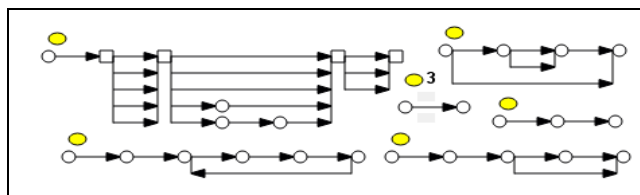
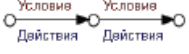


Рис. 3. Р*- схема программы без деталей реализации **компактнее в 45.1** раза.

качестве Условия записывается всегда истинная логическая константа – ключевое слово (**TextFile, array of string, Boolean** и т.д.), которое определяет Действия - структуру данных под дугой. В третьей колонке описаны МЕТОДЫ, в четвертой - СВОЙСТВА и МЕТОДЫ для свойств. Итого задаются семь методов, которые определяются далее другими семью Р*-схемами на рис.3. Из этих методов три задаются простейшей одной дугой.

В программе на рис.3 есть одна особенность, которой нет в традиционном программировании – это возможность очень просто задать специальное использование и выполнение дуг. Для этого достаточно двойным кликом мыши изменить конфигурацию вершин. На рис.3 изменить на квадратик – и в результате в одну колонку записываются безальтернативные дуги эквивалентные следующей последовательной

записи дуг:  и т.д. Это позволяет увеличить наглядность и компактность записи описаний данных. Аналогично также просто можно задать параллельное выполнение дуг, ожидание внешнего прерывания, запрет или разрешение прерывания и т.д. В третьей колонке продемонстрирована удобная возможность совмещения описания с определением, чего так же нет в традиционном программировании.

Запись программы на рис.3 традиционным способом в языке **Delphi** с использованием операторов **if, for, goto** и т.д. занимает **2.5 страницы** текста [8]. Р-схема этой программы компактнее в **5.3 раза**, а Р*-схема компактнее в **45.1 раз**. Из программы в **Delphi** удалено **860 (41.3%)** ненужных для Р-схемы символов.

Это значит во столько раз увеличивается обзорность, понятность, наглядность,

человечность программы в Р-схемах и во столько раз упрощается и повышается качество процесса ее изготовления. Программирование стало доступно ВСЕМ не только программистам.

РР*-схемы рис.3 впервые (чего нет в традиционном программировании) являются не только графической оболочкой программы, но и сетевым графиком ее разработки. По нему, по первой Р-схеме рис.3 автоматически генерируется план и порядок определения всех семи ООП-методов. Р-схема (нагруженный по дугам граф) – это универсальный способ задания информации в математике. Исторически такие графы (схемы) давно используются в *Сетевом планировании и управлении*, обеспечивая наиболее наглядное представление о ходе работ, их обеспеченности и завершенности, см. Рис.4. На дуге сверху таких графов обычно записывается название работ: 01, 11, 12 и т.д., а снизу или в скобках (см. первую дугу Р-схемы, Рис.4) – время их выполнения. Использование таких схем в программировании открывает большие перспективы организации планируемой индустрии разработки программных проектов в единой системе обозначений: пожеланий Заказчика, объектов Программы и системы Управления их разработкой.

III. ТРИ ОСНОВНЫХ ПРЕИМУЩЕСТВА НОВОЙ ТЕХНОЛОГИИ

1. Простота и несравненная наглядность. Важной особенностью ВТР, обеспечившей ей простоту и несравненную наглядность, является **исключение** из программирования операторов **if, for, goto** и т.д. (**всех!** около 10), которые оставались в программировании неизменными с конца 40-х годов. Эти машинно-ориентированные операторы и сопутствующие им символы-паразиты (**then, else, do** «;» и т.д.), **метки**, скобки типа **begin-end, {-}** и т.д. запутывают в общем-то простой, ясный и понятный процесс программирования, являются источником всех его сложностей и проблем, когда в результате «за деревьями – леса не видно».

Вместо них вводится ОДНА более мощная и всем понятная с одного взгляда, человеческая

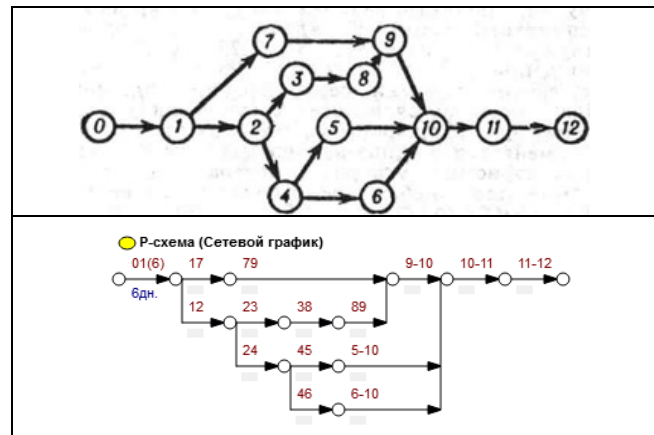


Рис. 4. Запись сетевого графика в Р-схемах.

(не машинная) сущность (рис.1), которая отражает процесс мышления самого человека (документирует процесс его мышления) и подключает к программированию более мощный зрительный аппарат ассоциативного мышления человека. Это снимает многие проблемы современных программистов, открывает дорогу для их объединения и накопления профессионального опыта. Р-схема программы впервые совпадает с сетевым графиком ее разработки и автоматически генерирует план работы коллектива программистов.

2. Компактность до 50 и более раз по сравнению с традиционной записью программ в языках программирования. Это самая компактная форма записи программ из всех известных. Она позволяет наглядно задать любую иерархию (вложенность) определений на естественном для каждого этапа разработки языке. При этом важно заметить, что вложенность типа вложенности операторов в традиционных языках **исключена**, а это сильно упрощает понимание программы в Р-схемах. Главное достоинство новой графической Р*-формы, которой **НЕТ** в традиционном программировании, – сразу видна суть, идея, схема всего проекта, по которой удобно понимать (объяснять, обсуждать, защищать, развивать, дополнять, утверждать, пересылать и т.д.) программный проект, а это оказалось очень важным в профессиональном программировании. Компактность РР*-схем тем выше, чем больше программный проект.

3. Мощьность. Приведенной одной дуги Р-схемы достаточно, чтобы записать любой алгоритм. Практически любой язык (включая естественный) может иметь указанную простейшую графическую оболочку, единую(!) для всех языков. Это значит все, что может быть записано на этих языках, может быть ЛЕГКО записано в графической их оболочке, которая эффективнее (значит компактнее, проще переводится в язык ЭВМ и имеет лучшие характеристики результирующего кода ЭВМ) традиционной их записи в виде текстов. Обратное много сложнее или вообще непонятно пока как потому, что Р-схемы мощнее и многое задают впервые, чего (эквивалента) НЕТ в традиционном программировании. То, что с помощью Р-схем впервые можно записать логику естественных текстов по новому решает вечно неразрешимую проблему ЗАКАЗЧИК-ПРОГРАММИСТ и обеспечивает впервые Доеазательство правильности программ.

В новой графической парадигме, инструмент (Р-схема) гибко подстраивается, развивается под решаемую задачу и ее исполнителей, а не задача трансформируется под существующие фиксированные операторы языков (инструмент), как сейчас в традиционном программировании. В результате сейчас программист (человек) вместо решения своей задачи вынужден подстраивать ее решение и свои мысли под эти машинные операторы, которые не имеют никакого отношения ни к человеку, ни к его задаче и совершенно далеки и не приспособлены для решения задач человеком – они слишком сложны и примитивны (маломощны, корявы и габаритны).

В технологии ВТР очень многое начинается со слов ВПЕРВЫЕ и В ОТЛИЧИЕ ОТ традиционной технологии программирования. Например: отображение сути Р* программы (алгоритма) без деталей реализации, которая компактнее в 50 и более раз традиционной записи программ; самодокументируемость программ, отсутствие вложения операторов, самая компактная запись и быстрый ввод программ из всех известных, доказательство правильности программ, трехмерное и

многомерное программирование, простое и наглядное задание параллельного выполнения программ, метаязык для формального задания синтаксиса и семантики языков, генерация (автоматическая) плана работы коллектива программистов и т.д. и т.д. Это только вершина айсберга преимуществ.

IV. СРАВНЕНИЕ С ДРУГИМИ ГРАФИЧЕСКИМИ МЕТОДАМИ ПРОГРАММИРОВАНИЯ

В настоящее время широкую известность имеют Графы, нагруженные по вершинам и Визуальное программирование. Широкое использование в программировании Графов, нагруженных по дугам неизвестно [9-11].

По нашему мнению, цель (назначение) первых (всем известных блок-схем, UML-диаграм, ДРАКОН-системы и т.д. более 100 наименований) заключается в том, чтобы хоть как-то сгладить, нейтрализовать недостатки (убожество) текстовых (античеловеческих) операторов программирования ЭВМ. В классической математике (теории программирования) общеизвестно, что такие графы не эффективны для использования в программировании и используются лишь для моделирования, проектирования и т.п. на начальных этапах поиска (конструирования) алгоритма, чтобы хоть как-то очеловечить хотябы первый контакт с ЭВМ. По сравнению с одной дугой Р-схемы они существенно более громоздки и не применимы на всем жизненном цикле программ.

Опять же по классической математике известно, что нагруженные по ДУГАМ графы (которыми являются Р-схемы) используются для построения (исследования) систем искусственного интеллекта (интеллектуальных систем), которые ближе к сути труда программистов и как показал наш небольшой опыт ИДЕАЛЬНО подходят для автоматизации (и очеловечивания) этого труда.

В настоящее время распространение получил термин **Визуальное программирование** как способ создания программы для ЭВМ путём манипулирования графическими объектами (иконками, кнопками, ярлыками, логоблоками и т.д.) вместо написания её текста. В последнее время визуальному программированию стали уделять больше

внимания - в связи с развитием мобильных сенсорных устройств (КПК, планшеты), когда использование клавиатуры не очень удобно. Сам термин Визуальное программирование всеми понимается гораздо шире, чем он трактуется его практикующими специалистами. В Диссертации Jeffrey Nickerson «Visual Programming» #9514409 проведен всесторонний анализ этому направлению. Главный вывод этой диссертации: **«Визуальные языки программирования не практичны, текстовые представления являются более эффективными и компактными, будущее за гибридными системами».**

Для нас этот вывод означает, что НАШЕ визуальное программирование (P-схемами) не сопоставимо **лучше** по сравнению с тем, что сейчас есть и принципиально, намного превосходят и визуальное и текстовое представления и по эффективности и по компактности **в 50 и более раз.**

ЗАКЛЮЧЕНИЕ

Таким образом, P-схемы – это универсальная графическая оболочка одна для всех языков, включая естественные и все языки программирования, которая сформировалась в результате многолетнего анализа существа и выявления недостатков организации традиционного процесса программирования. В результате программирование из примитивно-ремесленного стало математическим, человеческим и получило возможность развиваться по многовековым принципам математики. При этом важно, что новая графическая концепция сохраняет преемственность с тем, что уже есть и не отвергает все ранее достигнутое и запрограммированное, облекая их в единую и

более эффективную графическую оболочку, которая своей простотой обеспечивает возможность программировать ВСЕМ не только программистам. Все сделано по А.Эйнштейну: «делай настолько просто, насколько возможно, но не проще этого», потому что проще и эффективнее этого ничего пока нет.

REFERENCES

- [1] V.M. Glushkov and I.V. Velbitskiy, "Programming technology and problems of its automation", *USIM*, Kyiv, no. 6, pp. 75-93, 1976.
- [2] I.V. Velbitskiy, "Programming technology", *Technika*, Kyiv, p. 279, 1984.
- [3] *Information technology, Program constructs and convention for their Representation*, International standard ISO/IEC 8631, Second edition 1989 Geneva 20, Switzerland, ISO/IEC Copyright Office, p.7, 1989.
- [4] W.K. McHenry. "Technology: A soviet visual programming", *Journal of Visual Languages and Computing*, vol.1, no. 2, pp. 199-212, 1990.
- [5] L.F. Drobushovich, "Common use of UML and R-chart notations in the training process for software system development methods", *MEDIAS-2010*, Cyprus, pp.73-77, 2010.
- [6] I.V. Velbitskiy, "Next generation visual programming technology with R-charts", *Plenary report, MEDIAS-2012. Dedicated to 100 anniversary of Alan Turing (IEEE)*, Cyprus, pp. xiv-xxxiv, 2012.
- [7] I.V. Velbitskiy, "Graphical Programming and Program Correctness Proof", *Plenary report, 9th IEEE Computer Science & Information Technologies Conference*, Yerevan, Armenia, 23-27 Sept.2013, IEEEExplore,CSIT-20.
- [8] A.N. Valvachey etc. "Programming in Delphi. Textbook. Chapter 3. Object-oriented programming", *INTERNET*, p. 19, 2005.
- [9] Leonid Cherniak, "SOA – step beyond the horizon. Open systems: [IT management](#)", *Open systems #09/2013*, p.12
- [10] Графическое программирование на базе нагруженных по вершинам графов. http://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%B7%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5
- [11] Диссертации Jeffrey Nickerson «**Visual Programming**»#9514409 <http://c2.com/cgi/wiki?GraphicalProgrammingLanguage>