

Жаринов В.Н.

ПРЕДЛОЖЕНИЯ ПО РАЗВИТИЮ МЕТОДОЛОГИИ ДРАКОН

Обзор и анализ

Версия 05.04.10

ВВЕДЕНИЕ В ДОКУМЕНТ

Общие положения

1. Файл содержит выполняемый автоматизированным способом (в форме машинного оригинала МО) <беловик|черновик> целевого документа или его части (неотъемлемой), выделенной для удобства работы.

Документ в целом, кроме основного содержания, может включать приложения. Содержание документа, приложения (его выделенной части) составляют текст и/или иллюстрации (графчасть).

Конкретное наполнение файла определяется по его имени (полный формат имён см. шаблон документа)¹.

2. Содержание документа, приложения подразделено на структурные элементы по иерархии; её высшие 4 уровня стандартны. Элементы обычно имеют многоуровневую нумерацию и заголовки-абзацы, входящие в оглавление; возможны также элементы без нумерации, в т.ч. не входящие в оглавление, в т.ч. с заголовками в тексте.

В тексте применяются типовые приемы оформления, описанные в п/р 1.1 документа|шаблона.

3. В файл части из документа, приложения выделяется элемент структуры стандартного уровня иерархии (или ряд соседних элементов одного уровня) целиком (с заголовками).

Для многофайлового МО в имени каждого файла указаны индексы входящих элементов (формат: разделы <CH>, подразделы <PNN>, пункты <PINNN>, подпункты <PINNNN>); файл первой части является *головным*.

При наличии приложений их форму (способ выполнения) указывают в отметках о наличии в составе единственного (или головного) файла основного документа (виды способов и формат отметок см. шаблон).

Приложения в МО могут выполняться как отдельные файлы *ПрилN* (что указывается в их отметках о наличии).

При наличии иллюстраций в документе, приложении (части) они также м.б. выполнены разными способами. Подрисовочные подписи включаются в оглавление для удобства поиска рисунков в документе.

Иллюстрации в МО могут содержаться в отдельном файле графчасти *Рисунки*; тогда текст содержится в файле *Текст*, и в нём дублируется подпись к каждой иллюстрации по месту её упоминания для отсылки к графчасти.

4. Оригинал документа (части) выполнен как настоящий файл (имя см. поле внизу) и другие необходимые (детальный состав многофайлового документа см. п. 1.1.4 в <настоящем файле|головном файле *Ч.1 Введ.*>).

Текст подготовлен в среде OpenOffice.org Writer или иной программы, совместимой по файлам; иллюстрации выполнены в той же программе и/или иными средствами, включая захват машобrazов для МО.

Подлинник выполняется как твёрдая копия с заменой и/или добавлением листов к твёрдой копии предыдущих версий, либо как машинный образ файлов оригинала по листам, с которого делаются твёрдые дубликаты.

5. Все права защищены их обладателями. Документ, а равно любая его часть в любой форме адресованы лицам, которые указаны автором как его адресаты и (или) третьим лицам, участвующим в совместной деятельности по соглашению между автором и указанными лицами; иное возможно только с письменного разрешения автора.

Документ предназначен для учебных, информационных, научных или культурных целей в соответствии с действующим законодательством РФ, включая, но не ограничиваясь, п.1 Ст.1274 ч.4 ГК РФ². Содержание документа используется «как есть», без к.-л. изменений. ПОЛЬЗОВАТЕЛЮ РАЗРЕШАЕТСЯ: создать резервную копию каждого файла оригинала (при предоставлении только подлинника – каждого его листа) на случай утраты; делать одну твёрдую копию МО для правомерного пользования, включая замену утраченных (испорченных, потерянных) листов; цитировать документ в объёмах и порядке, разрешённых нормами авторского права РФ. ПОЛЬЗОВАТЕЛЬ ОБЯЗАН: использовать оригинал (подлинник) и его копии (резервную и/или твёрдую) только лично и как указано выше; при цитировании документа ссылаться на источник³. Иное воспроизведение документа или любой его части невозможно без письменного разрешения.

Информация, содержащаяся в документе, получена из открытых источников, рассматриваемых автором как надёжные. Возможное наличие секретных, конфиденциальных, а равно иных сведений ограниченного доступа следует рассматривать как результат предположения на массивах открытых сведений. Имея в виду возможные человеческие и технические ошибки, автор не может гарантировать абсолютную точность и полноту приводимых сведений, и не несёт ответственности за возможные последствия, связанные с их использованием.

¹ Переменные части текста даются как поля в "< >", заменяемые на описание; общая часть (корень) поля пишется как есть, а изменяемые части как "*". Файлы МО с однокоренным именем относятся к одному элементу структуры.

² Федеральный закон № 230-ФЗ от 18 декабря 2006 г.

³ Если цитата состоит полностью из сведений, цитирующих другой источник – дать ссылку на первоисточник.

Назначение, сведения о версиях, языковые соглашения

1. Документ содержит выборку из обзора и анализа различных предложений.
2. Версии документа выпускаются по мере обновления содержания, входящего в выборку.
3. В тексте употребляются следующие типовые обозначения и сокращения:

англ.	английский;
букв.	буквально;
в т.ч.	в том числе;
и т.д.	и так далее;
и т.п.	и тому подобное;
к.-л.	какой-либо;
напр.	например;
см.	смотри;
т.е.	то есть;
т. зр.	точка зрения;
т.о.	таким образом;
разд.	раздел (документа);
п/р	подраздел (документа);
п.	пункт (документа);
п/п	подпункт (документа);
пред.	предыдущий;
след.	следующий;
ТЛ	титальный лист;

Сведения о терминологии, обозначениях и сокращениях данного документа см. в п/р 1.3.

Оглавление

1. ВВЕДЕНИЕ	4
1.1. Общие указания	4
<i>[Подпись к иллюстрации (абзац уровня 5)]</i>	5
1.2. Общие положения	5
1.3. Необходимые определения	5
1.3.1. Текстовая часть	5
1.3.1.1. Условные обозначения	5
1.3.1.2. Основные термины документа	5
1.3.1.3. Сокращения	5
1.3.2. Графическая часть	5
2. ПРЕДЛОЖЕНИЯ СООБЩЕСТВА ПО МЕТОДОЛОГИИ ДРАКОН	6
2.1. Предложения Д.В. Барановского	6
2.1.1. О целевом назначении предложений	6
2.1.2. Конкретные положения	7
2.1.2.1. О многоуровневости редактирования модели	7
2.1.2.2. О соотношении техноязыка и ВЯЗБС	7
2.1.2.3. О графике икон в ВЯЗБС	8
2.1.2.4. О правилах линейного отображения ТФАП	8
2.1.2.5. О полноценности среды программирования	9

2.1.2.6. Вкратце об остальном :)	10
2.1.3. Итоговое мнение	10
5. ИСТОЧНИКИ	12
Документы для ссылок	12
Полезные ресурсы	12

ПРИЛОЖЕНИЯ:

—резервная страница оглавления—

1. ВВЕДЕНИЕ

|| Внимание: следует хорошо изучить этот раздел, чтобы ориентироваться в документе.

1.1. Общие указания

1.1.1. В **тексте** документа применяются следующие приемы оформления.

1.1.1.1. Чтобы **упорядочить работу с материалом**, каждая новая мысль начинается с красной строки, а кроме того, для наглядности часть абзацев имеет особые стили:

- элементы перечисления удобно оформлять как пункты маркированного списка;

Без красной строки оформляются абзацы, отбиваемые в объёмном тексте мысли для удобства чтения (за первым), а также вводные положения к крупному элементу (под заголовком).

Формулы обычным текстом даются центрованно отдельных строках

Таковыми абзацами оформлены мысли, на которые Вам следует обратить особое внимание (узловые моменты текущего пункта или наиболее важные выводы из него).

Внимание: [особое указание, требование, необходимое условие для применения пункта]

- так выделяется пункт перечня, требующий особого внимания;

Так выделяется формула обычным текстом, требующая особого внимания

Эти сведения даются сразу перед тем (после того) основным текстом, к которому относятся.

Пример. Таковыми абзацами выделяются примеры, иллюстрирующие текущую мысль основного текста. Так же выделяются практические рекомендации, советы, указания.

- так в тексте примера выделяется пункт перечня;

Так в тексте примера оформляется абзац продолжения текущей мысли.

Так выделяется формула обычным текстом в примере, примечании

Абзац с отступом и уменьшенным шрифтом выделяет в тексте документа составляющие развития, которые дополняют (уточняют, конкретизируют) содержание основного текста.

- так в тексте развития выделяется пункт перечня;

Так в тексте развития оформляется абзац продолжения текущей мысли.

Так в тексте развития выделяется формула

Информация к размышлению. [подзаголовок статьи]. Таким образом в тексте выделена познавательная составляющая, которая не является обязательной для изучения, но может расширить и углубить понимание предмета.

- так в тексте отступления выделяется пункт перечня;

Так в тексте отступления оформляется абзац продолжения текущей мысли.

1.1.1.2. С той же целью фрагменты в тексте могут оформляться в следующих стилях:

Жирным шрифтом выделены названия отдельных пунктов, уровни классификации или комментарии в тексте к элементам схем, диаграмм.

Курсивом выделяются понятия, определяемые в тексте документа, а также предложения, содержащие важную информацию (выводы, указания и пр.). В списке литературы курсивом выделены позиции, которые имеются в [учебной|служебной] библиотеке.

Жирным курсивом выделены подуровни классификации либо понятия, о которых идет речь в окружающем тексте, или которые уже должны быть Вам известны.

Подчеркиванием обозначаются ссылки на место в данном документе или за его пределами, напр., на другие документы (кроме гиперссылок на ресурсы интернет, которые оформляются стандартно для инфордоков). Если подчеркнута ссылка на другие дисциплины, сферы деятельности, то Вы можете обратиться за информацией к соответствующим специалистам, преподавателям, в интернет.

Разрядкой выделены места, на которые следует обратить особое внимание.

Таким начертанием (гарнитурой) шрифта и курсивом выделены наименования объектов (сущностей), описываемых в документе.

Такой гарнитурой шрифта (с уплотнением) выделены тексты процессов (алгоритмов, программ).

Так выделяется внутритекстовый заголовок части пункта (подпункта). Далее начинается собственно текст этой части. Такой заголовок не входит в оглавление документа.

1.1.2. В **графической части** документа используются стандартные стили текста и условные обозначения, приведённые далее (см. п. 1.3.1).

Иллюстрации упорядочены по тексту и снабжены подписями вида:

[Подпись к иллюстрации (абзац уровня 5)]

В файле выделенного текста эти подписи указывают наличие и положение иллюстраций.

Внимание: отдельные подписи могут размещаться не под, а над рисунком. В любом случае подпись относится к тому рисунку, к краю которого она расположена вплотную.

Иллюстрации оформлены с применением метода КогниСтиль (см. /1, Гл.19/ и п.1.4.3).

1.2. Общие положения

1.2.1. В документе сжато излагается авторская т. зр. о предложениях по развитию (модификации, расширению) современной отечественной методологии ДРАКОН, предназначенной для описания человеком чьей-либо (или собственной) деятельности, т.е. **формализации** (или **автоформализации**) **технологических знаний**⁴.

Развёрнуто методология изложена в книге её создателя /1/.

1.2.2. Документ предназначен для сведения о представлении на базе ДРАКОНа **формализуемых** (в смысле /3, п. 1.4.1/) знаний, в первую очередь о содержании (логике) трудовых (бизнес-) процессов, в графической (схемной) форме в целях инжиниринга (реинжиниринга) бизнеса, управления качеством (в т.ч. в рамках СМК ИСО900N).

Документ адресован носителям знаний о процессах в организациях различных сфер деятельности, а также специалистам по формализации указанных знаний.

1.2.3. Документ подготовлен с использованием источников информации, указанных в разд. 5. Необходимая информация содержится также в приложениях к документу:

1.2.4. Основное содержание документа в оригинале выполнено как настоящий файл.

Приложения к документу выполнены как отдельные файлы (см. отметки о наличии).

1.3. Необходимые определения

1.3.1. Текстовая часть

1.3.1.1. УСЛОВНЫЕ ОБОЗНАЧЕНИЯ

Основные группы текстовых обозначений, используемых в данном документе, даны в /3, п. 1.3.1/.

1.3.1.2. ОСНОВНЫЕ ТЕРМИНЫ ДОКУМЕНТА

Специальную часть словаря документа, непосредственно связанную со свойствами дракон-схем («диоформа», «лиоформа» и т.п.) см. в /3, п/п 1.4.2.2; 1.4.2.3/. Употребление этих терминов см. напр. в /3, п/п 2.2.1.1/, /3, п. 2.2.2/, /3, п/п 2.3.1.1/.

Основную часть словаря см. в /3, п. 1.3.2/. Терминологию ДРАКОНа и программирования дополнительно см. в /Драконографика, Стр. Легкомысленный словарь; Более строгий словарь/.

1.3.1.3. СОКРАЩЕНИЯ

Сокращения, употребляемые лишь в отдельных местах текста, расшифровываются там же. Если различные сокращения обозначают одно и то же (напр., на разных языках), то их расшифровки содержат отсылки друг к другу.

1.3.2. Графическая часть

Графические обозначения употребляются в смысле, определённом в /3, п/п 1.2/.

⁴ Данные понятия объясняются далее в п.1.4.1, а также в работах /1, Гл.3/ и /2, Разд.6/.

2. ПРЕДЛОЖЕНИЯ СООБЩЕСТВА ПО МЕТОДОЛОГИИ ДРАКОН

Здесь даётся авторский обзор и анализ предложений по развитию (модификации) языка, технологии и/или реализации МФЗ-ДРАКОН, как правило, от профессиональных участников дракон-сообщества, представленного на веб-форуме /4/.

2.1. Предложения Д.В. Барановского

2.1.1. О целевом назначении предложений

В явном компактном виде не нашёл, поэтому сформулирую так, как сам понял.

2.1.1.1. Исходная задача, решаемая Дмитрием – позволить разработчику работать с моделью задачи (визуалом или дракон-моделью процесса решения) на всех этапах существования символического (воспринимаемого человеком) представления решения – вплоть до исходного текста программы (символ-импер-текста).

Для реализации данной цели вводится предметное представление структуры алгоритма в виде человекочитаемого текстового файла; автор использует формат, называемый ТФАП.

Предлагается изображать алгоритм, описанный в ТФАП, в виде псевдографической блок-схемы (ПГБС), как показано на Рис. 6 Предложения,⁵ по сути, это лиоформа визуала, т.е. результат выкладки его граф-схемы.

Маршрутный подязык ПГБС – сверхминимальный базис блок-схем (где не только оставлены лишь два типа операторных вершин – функциональные и предикатные – но и формы этих вершин унифицированы как прямоугольники, только с разным числом выходов). Это следствие того, что для построения икон и линий используются только знаки псевдографики алфавитно-цифрового дисплея (АЦД), рассчитанные лишь на построение прямоугольных контуров.

Генерация кода (исходного текста) по ТФАП предполагается предопределённой (нет речи о программных приложениях икон, редактируемых оператором), но нежёстко заданной – лексика языка исходного текста определяется в словаре (т.н. секции задания параметров).

Из примера секции в Разд.6 можно понять, что командный подязык текста икон – вариант русского псевдокода.

Указанные соображения воплощены Барановским в среду разработки и документирования программ (далее – Б-среда). В ней возможны два вида оболочек для работы с ТФАП:

- полнографическая – отображает ТФАП как граф-схему в диоформе (как у обычного дракон-редактора) и графически;
- псевдографическая – отображает ТФАП в лиоформе (с выкладкой дракон-схемы) и только знаками псевдографики, стандартными для АЦД информашин типа ПК ИБМ.

По сути, имеем базис представлений размерности 2х2 (по отображению – графика/псевдографика; по организации – диоформа/лиоформа); из 4-х вариантов прямого произведения использовано 2.

Функции редактирования, как я понимаю, в обеих оболочках одинаковы. Изначально реализована псевдографическая оболочка со всеми вытекающими требованиями к символическому представлению блок-схем.

2.1.1.2. Главное ЦН рассматриваемого предложения – использовать опыт создания Б-среды для реализации ДРАКОНа. При этом:

- для визуалов также вводится визуализация в лиоформе и псевдографическое отображение (далее – ПГО) наряду с графической дракон-схемой (ГДС);
- дракон-схемы(-модели) также представляются как ТФАП-файлы.

Из этого вытекают конкретные представления Дмитрия об изменении ДРАКОН-методологии (языка, технологии, реализации), к которым далее и перехожу.

⁵ Далее все специально не уточнённые ссылки относятся к предложениям Д.В. Барановского.

2.1.2. Конкретные положения

2.1.2.1. О многоуровневости редактирования модели

В п. 2.1.2 утверждается: «разработки и документирования ПО с графическим интерфейсом – одновременно доступны как текстовое окно редактирования исходного кода ПО, так и графическое окно просмотра и редактирования блок-схем алгоритмов работы ПО». В связи с этим возникает вопрос – как согласовывать изменения, вносимые на уровнях текста и схем?

Можно действительно объявить истинно предметным представлением файл исходного текста и тогда допустить при его редактировании перестроение ТФАП-файла с переотображением последнего в лио- и диоформе. Правильно я понимаю Вашу мысль?

В принципе логично будет и считать истинно предметным представлением модели ТФАП-файл, по которому в полноценной среде строятся одновременно различные его символические представления – линейная выкладка ТФАП-файла (как на Рис. 6) и планарная дракон-схема.

В любом случае работа на разных уровнях м.б. только последовательной: отредактировал оператор среды сам ТФАП-файл – в него вносятся одни изменения, отредактировал дракон-схему – опять же в ТФАП-файл вносятся другие (с отверганием недопустимых, разумеется); и тут же по принятым изменениям перестраиваются и ГДС, и ЛО.

Кроме того, истинным м.б. только одно представление модели – аналогично принципу единства базы данных. Нужно определиться – будет ли это файл ТФАП или исходного текста? Мне представляется предпочтительным первое – из одного ТФАП можно получать разные исходные тексты для разных языков (диалектов).

Аналогично добавляется редактирование ПГО-схем; однако сами эти схемы считаю применительно к ДРАКОНу неоправданными, о чём ниже.

2.1.2.2. О соотношении техноязыка и ВЯЗБС

Основные соображения по данному поводу приведены в п/р 4.1 как нумерованные пункты-недостатки; разберём каждый из них.

Недостаток 1) в техноязыке снят путём подстановки (БП с возвратом); сам же Дмитрий в 3) говорит о том же как о ссылке одного алгоритма на другой. Поэтому мы должны не просто «часть ПГБС, в случае нехватки места, перенести на следующую страницу», а оформить эту часть как вставку (подпрограмму, процедуру), т.е. делать то же, что Дмитрий предлагает для случая 2).

В отношении 3) – не совсем понятно, почему наглядность ссылки по индексу выше, чем по имени визуала в иконе Вставка. В п/р 4.2 фактически ничего, кроме «нижнего этажа» в блоках действия для записи ссылок, я не увидел. Или Дмитрий имел в виду что-то иное?

По поводу 4) – визуализация подразумевает, что программист не «отвлекается на рисование» а рисует алгоритм, а затем, как Паронджанов говорит, описывая работу в ГРАФИТ-ФЛОКС, «нажимает кнопку и получает файл исходного текста».

Давайте разделять процессы спецификации программы (результат – исходный чертёж, представленный, допустим, в ТФАП) и кодогенерации (результат – машинный код ТФАП-файла/файлов, а до этого – исходный текст) и соответственно определяться, какой тип процессов считаем «творческим», а какой «рутинным».

Концепция ДРАКОНа исходит из того, что сочинение исходного чертежа – это творческая деятельность, поручаемая человеку; у Дмитрия же, похоже, получается наоборот. В то же время можно сделать и тот, и другой тип процессов нерутинными; но для этого надо не отказываться от положений шампур-метода, а творчески развивать их.

Однако претензия: «...налицо трудоёмкость рисования и исправления блок-схемы.» справедлива, если среда визуализации предоставляет сочинителю только строгий (автоматический, «цербер-») режим, о чём подробнее см. п/п 2.1.2.5 данного документа. Это до последнего времени было характерно для среды Г.Н. Тышова (см. /2/ и обсуждение на /4/).

2.1.2.3. О графике икон в ВЯЗБС

Теперь к положениям Разд.7. Представляется неоправданной унификация всех форм икон; даже в машинном языке имеем разные типы команд процессора (общеизвестное подразделение: действия, пересылки, передачи управления, специальные) – что уж говорить о более высокоуровневых языках.

О форме иконы Вопрос: повлиявшие на автора «неформальные» соображения тех или иных абонентов интернет м.б. вполне преднамеренными (как попытка «упаковать в непривлекательный фантик» непонятую/конкурирующую идею для манипуляции сознанием в полном соответствии со сказанным, напр. у того же Почепцова); в этом случае расчёт как раз на такую реакцию (участники дракон-сообщества будут пытаться как-то изменять язык, не обязательно в лучшую сторону, а там можно ещё какие-то претензии придумать – от всех неприятных ассоциаций, навязанных изобретательным умом, не избавишься :)). Если же это не преднамеренная позиция, а просто стёб, то тем более вряд ли нужно придавать ему такую важность – давайте всё-таки исходить каждый из своих представлений.

Отсюда вопрос к автору: лично у Вас есть содержательные доводы за отказ от формы иконы Вопрос и остальных икон?

Ответ я вижу один – несмотря на развитость алфавита ДРАКОНа, приложить к нему принцип ПГО, реализованный для ВЯЗБС. Чисто технические оправдания этому можно, видимо, найти – скажем, по согласовании открытого формата ТФАП-файла фактически можно вести обработку принципов среды визуализации на псевдографической основе; я так понимаю, ПГО-приложения будут получаться менее «тяжёлыми», легко модифицируемыми, и это позволит наращивать их функциональность легче, быстрее и разными разработчиками; впоследствии каждый из них будет выпускать полнографическую среду, для которой псевдографическая послужит своего рода прототипом.

Жертвовать во имя этих (или иных) соображений когнитивным качеством языка (сводя его к БС в «хуже, чем минимальном» базисе) считаю неоправданным. Да и у самого Дмитрия: «для отображения БС нужно использовать полноценную векторную графику, что позволило бы не только повысить качество графического отображения блок-схем, но и без особых усилий, просто отсканировав графические координаты элементов блок-схем, выполнять генерацию dxf-файлов для системы AutoCad.»

Аналогично не могу согласиться с определением атрибутов графики и текста икон как с «фокусами» – это нужно для человека-читателя схем. И опять процитирую Дмитрия: «...важным сейчас является вопрос, как должны выглядеть отображаемые новой графической оболочкой блок-схемы – и здесь уже возможны варианты.»

Вопрос: под вариантами понимается отображение наряду с диоформой также лиоформы визуала? или речь идёт о ревизии правил построения икон и/или дракон-схем?

Как я понимаю, алфавит и правила техноязыка выработаны в т.ч. с использованием методов инженерной психологии, на основе экспериментальной отработки лучшего восприятия дракон-схем. И менять их можно только на научной основе.

2.1.2.4. О ПРАВИЛАХ ЛИНЕЙНОГО ОТОБРАЖЕНИЯ ТФАП

ВЯЗБС-силуэт, показанный на Рис. 5, следует считать промежуточным результатом выкладки в лиоформу – обозначения веток устранены за счёт «вноса» переходов в текст икон, но форма силуэта ещё сохраняется. Предполагается, что так понятнее; однако мне понятнее либо нормальный силуэт, либо конечный результат на Рис. 6, где не надо извлекать смысл переходов по петле силуэта из икон где-то внутри тела.

Считаю необходимым для сохранения эргономичности в ЛО ТФАП сохранить принцип различения условных и безусловных переходов: вертикали первых упорядочиваются вправо от шампура, вторых – влево (уже не сливаясь в петлю разрешённого БП, как в силуэте).

На мой взгляд, необходимо использовать, с одной стороны оригинальные диоформы шампур-метода – силуэт и примитив, а с другой – результат их выкладки – логически одномерную схему с БП-соединителями вертикалей (единообразную для обеих диоформ).

В то же время и такой форме схем можно найти применение. В самом деле, ВЯЗБС-силуэт по сути есть аналог схемы подключений блоков оборудования; форма иконы для этого назначения не важна, а типы блоков можно указывать их текстом. Видимо, исходя из того, что цепь между импер-блоками всегда единственная (переход по адресу), Дмитрий счёл возможным не рисовать «разъём», а просто ввести в контур блока поле адреса подключения.

Однако, если оборудование как физическую систему нужно монтировать при участии людей, то схему программы как знаковую систему можно и нужно «монтировать» автоматически, что и делает транслятор. Для чего ещё может пригодиться такая схема? Пожалуй, для отладки дракон-программы — как и любая монтажная схема. Возможное применение в этом качестве обсуждается в п/п 2.1.2.5.

2.1.2.5. О полноценности среды программирования

В п. 2.1.1 утверждается: «полноценную интегрированную среду программирования (напр., MS Visual Studio)». Считаю, что полноценной как раз м.б. только среда гибридного программирования, сочетающая и когнитивно-эргономичное формализованное документирование, и управляемую (в т.ч. оптимизирующую) кодогенерацию по формализованным документам.

Любая система, где нет визуализации как алгоритмов, так и структур данных, не м.б. признана полноценной и интегрированной; если же мы хотим ввести ручную оптимизацию кода, то нужна ещё визуализация исполнения кода на реальной машине (лучшим решением представляется «визуальный отладчик» на базе метафоры «рабочей точки» Паронджанова).

Возможный вариант такого отладчика — с визуализацией движения рабочей точки по схеме типа ВЯЗБС-силуэта. Для текущего оператора фиксируется и отображается состояние системы момента исполнения (состояние программы); для остальных только отражается содержание (текст). Можно реализовать такую схему как для прогзыка исходного текста (иконе соответствует его оператор), так и для языка ассемблера (икона — машинная команда). При этом наложение состояния на схему кода служит эргономически более эффективной заменой его представления в пресловутом «главном окне сообщений WIntel»⁶. Конечно, нужно предоставить возможность ручной правки операторов и переменных состояния, установки контрольных точек на схеме. Вот и получится тот самый тестер для наладчиков программ :)

Что вытекает из такой прагматики? Прежде всего, нужно усовершенствовать описание «подключений» операторов; надо явно указывать адрес конца цепи (следующего оператора). Далее, всё-таки нужно ввести разную графику икон по типам операторов; если схема визуализирует код на ассемблере, то типов будет раз-два и обчёлся, а для восприятия удобнее. Наконец, нужно определить формат отображения переменных состояния (в основном внутри иконы).

В п/р 3.3 говорится: «1) Чрезмерная перегруженность всевозможными правилами ввода привела к тому, что написанная Г.Н. Тышовым и с.Дракон многими оценивается как неудобная в работе, отсутствуют гибкие инструменты быстрого внесения исправлений в ранее введённую визуальную схему;». Это очередное требование о необходимости нестрогого (полуавтоматического, «либерал-») режима визуализации; оно уже обсуждалось и по сути признано оправданным, однако нужно определиться — какие правила шампур-метода нужно отбросить в таком режиме? Хотелось бы услышать и мнение Дмитрия об этом.

В то же время соображение: «...использование для документирования ПО ПГБС, которые являются упрощёнными изображениями обычных блок-схем, не дает никакого выигрыша в смысле трудоемкости и лишь ухудшает читаемость таких блок-схем.» никак, на мой взгляд, не отменяется последующим: «...ситуация принципиально меняется при использовании простой программы, позволяющей генерировать файл ПГБС из текстового файла, содержащего текст программы или текст документации к этой программе. » — ПГБС, отображаемые исходя из возможностей контроллера АЦД для ПК ИБМ, не могу считать более эр-

⁶ Имеется в виду окно «программа выполнила недопустимую операцию и будет закрыта» :))

|| гономичными, чем текст в ТФАП-файле.

В п/р 3.3 говорится также: «В и.с.Дракон встроена возможность генерации программного кода для С, Паскаля и некоторых других языков программирования высокого уровня, но генерируется не полноценный исходный код, который сразу же можно было бы запускать на компиляцию, а некие шаблоны исходного кода, которые программист перед компиляцией должен еще дописывать.». Возможно, в первую очередь это следствие гибридизации сразу с несколькими проязыками по принципу «прямого произведения» их стандартов. Полагаю, в будущих средах визуализации сначала стоит реализовать один гибридный техноязык ДРАКОН-Х (напр., с Обероном-2), зато точно по спецификации Х. Для расширения же множества языков Х нужно определить «интерлингву», в которую переводить с любого из них; ситуация должна улучшиться.

2.1.2.6. ВКРАТЦЕ ОБ ОСТАЛЬНОМ :)⁷

Формат ТФАП-файла (Разд. 5) и генерацию исходного текста (Разд. 6) не обсуждаю – это вопросы во многом специальные. Отмечу лишь, что упрощение состава и структуры исходного текста действительно актуально.

Интересен обзор систем РДП в п/р 1.2. Понятно, что этих систем (в т.ч. для визуального/гибридного программирования) гораздо больше; достаточно посмотреть хотя бы журнал «Современные технологии автоматизации» за несколько последних лет. Большинство из них используют текстовые проязыки (по МЭК61131-3 или оригинальные), а среди визуальных – языки функциональных схем, но есть и блок-схемные. Кстати, в конце, видимо, очень показательный пример «костыльности» UML прямо «с поля битвы» :).

Описанная среда ATMEЛ вполне реализует «анархический» подход к схемам программ, описанный Паронджановым; вертикали в БС не упорядочены, графика вершин и линий определена лишь частично.

О том, «Почему развитие вроде бы нужной системы визуальной разработки ПО идет такими медленными темпами и не привлекает большого числа энтузиастов...?» (п/р 3.3). Один из ответов, на мой взгляд, косвенно даёт сам Дмитрий здесь: «Сегодня достаточно часто встречается ситуация, когда с уходом из организации старой команды программистов новой команде, приходящей ей на смену, приходится программировать программный проект, поддерживаемый организацией, практически заново.»(п/р 1.1). Пока не будет организационно-экономического согласования в процессе автоматизации (популярные тенденции с одной стороны – нанимать ИТ-специалистов «за медные деньги», с другой – иногда «набивать себе цену») – я думаю, не будет в ИТ-сфере заинтересованности в полноценном сопровождении проектов – это и определённая технологическая дисциплина (которую заказчик может требовать только от разработчика, не соблюдая сам), и риск «выкручивания рук» по принципу «раз всё документировано — разлука будет без печали». Конечно, самая совершенная РДП-среда здесь мало что изменит...

2.1.3. Итоговое мнение

Идея повышения «человекочитаемости» файла дракон-схемы представляется интересной, как и показ наряду с диоформой визуала его лиоформы (но также в полнографическом виде). Введение фактически двух методов визуализации – полнографического и псевдографического – исходя из вышесказанного считаю неоправданным.

Таковы пока соображения; возможно, они будут дополнены и/или уточнены впоследствии.

⁷ Такое подразделение использовано по аналогии с работой: Грызун В. Как Виктор Суворов сочинял историю. – М.:ОЛМА-Пресс, 2003.

[НАЗВАНИЕ]

В приложении кратко описаны .

5. ИСТОЧНИКИ

Документы для ссылок

1. Паронджанов В.Д. Как улучшить работу ума. Алгоритмы без программистов – это очень просто! – М.: Дело, 2001.
2. Тышов Г. Интегрированная среда ДРАКОН. Справка.
3. Жаринов В.Н. Описание деятельности на основе методологии ДРАКОН. Вводный цикл. – 2009.
4. [Визуальный язык ДРАКОН](#): веб-форум. – Конференция OberonCore.
5. Паронджанов В.Д. Занимательная информатика. – М.: Дрофа, 2007.

Полезные ресурсы

[Ты-среда](#) – здесь можно загрузить файлы установочного комплекта.

[Драконография](#) – данный веб-ресурс содержит извлечение из документа /3/ в объеме, достаточном для целей данного документа, и цитаты из некоторых других источников.