УДК 519.681

В.Г. Акуловский, А.Е. Дорошенко

Реализация комплексного подхода к описанию алгоритмов информационно-управляющих систем в рамках алгебраического аппарата

Показана возможность реализации в рамках алгебры алгоритмов с данными комплексного (от управления и данных) подхода к разработке алгоритмов информационно-управляющих систем. Такой подход может быть использован на любом этапе проектирования и для любой подсистемы, образующей систему.

An ability of realization within the framework of algebra of algorithms with data to cope with complex approach (including both control and data) to develop algorithms of the information/control systems is shown. This approach can be used at any stage of system development and any subsystem which constitutes the system.

Показано можливість реалізації в рамках алгебри алгоритмів з даними комплексного (від управління й даних) підходу до розробки алгоритмів інформаційно-керуючих систем. Цей підхід може бути використаний на будь-якому етапі проектування і для будь-якої підсистеми, що утворює систему.

Введение. Исторически сложились два подхода к программированию от управления (функционально-ориентированный подход) и от данных, которые нашли свое воплощение в виде структурной и объектно-ориентированной методологии программирования (ООП) [1, 2]. При всех положительных свойствах этих методологий они (как и соответствующие подходы к программированию) ориентированы на разработку различных классов программных систем. В частности, общепризнано, что первый эффективен при разработке систем автоматического управления (САУ), а второй – при разработке информационных систем, в частности автоматизированных систем управления (АСУ). В связи с этим выбор одной из методологий (одного из подходов) накладывает на процесс разработки ограничения, свойственные этой методологии. При этом обе названные методологии не формализованы в достаточной степени.

Очевидно, существуют классы систем, для которых ни одна из названных методологий в отдельности не есть оптимальной. Пример систем такого класса – информационно-управляющие системы (ИУС), о которых, исходя из многочисленных публикаций (например, [3]) и не взирая на различные классификации программных систем, можно сказать, что они сочетают функции АСУ с функциями САУ.

Ключевые слова: алгоритмы, алгебра алгоритмов, Д-операторы, информационно-управляющие системы, разработка алгоритмов.

Более того, даже в рамках системы, которую можно отнести к одному из этих классов, на различных этапах ее разработки возникают сложности, вызванные ограничениями выбранной методологии. Подтверждением тому служат работы, направленные на разработку новых методологий и использование мультипарадигменного программирования [4, 5].

Использование методологии проектирования алгоритмов, неадекватной конкретной решаемой задаче или ее подзадаче, – один из факторов, порождающих кризисные явления в программировании.

В связи с этим назрела необходимость в разработке формального аппарата, средствами которого был бы реализован комплексный подход к программированию, при котором, в процессе разработки программной системы, упомянутые подходы могли быть использованы не только по отдельности или совместно, но, в первую очередь, применяться для решения задач (подзадач, подсистем), входящих в разрабатываемую систему, специфике которых они адекватны.

Для построения такого формального аппарата была модернизирована [6] известная модель ЭВМ Глушкова.

Модификация заключается в дополнении модели внешней средой (ВС), состоящей из программной и аппаратной (АС) составляющих.

Первая обеспечивает взаимодействие между моделями, функционирующими в различных

ВС (многомашинная организация) и между задачами в рамках одной ВС (квазипараллельная, многозадачная организация). Вторая представляет собой память и внешние устройства (ВУ), включающие как устройства ввода–вывода (УВВ), так и устройства связи с объектом управления (УСО), и интерпретируется как данные, формализованные следующим образом [6].

Определение 1. Данными называется пара $D = \langle N, З \rangle$, где N – носитель данных, З – кортеж значений, носитель которых – N. На каждом шаге вычислительного процесса носитель содержит (хранит) некоторый (текущий) кортеж значений данных, в частности, эти значения могут быть неопределенными.

С учетом определения 1, АС рассматривается как множество данных $D^{\text{АС}} = \{\{D^{\text{П}}\}, \{D^{\text{ВУ}}\}\}$, где носителем $D^{\text{П}}$ есть память, а носителем $D^{\text{ВУ}}$ – ВУ. Эти данные характеризуются в каждый момент времени некоторым множеством значений, изменяющихся в ходе вычислительного процесса.

Такая модификация модели ЭВМ позволила ввести Д-операторы вида $(D)O(D')$ со специфицированными данными D и D', которые они обрабатывают, т.е. анализируют и преобразуют, изменяя их значения.

Заметим, что значения специфицируемых данных в общем случае изменяются в ходе вычислительного процесса, а на некоторых его этапах могут быть не определены. Отсюда следует, что специфицируются, фактически, носители данных. Поэтому под теоретико-множественными операциями, определенными на множествах данных, понимаются операции над их носителями.

Специфицируемые данные, с учетом сделанного замечания, такие, что $D \subseteq D^{\text{АС}}$ и $D' \subseteq D^{\text{АС}}$.

Отметим, что под данными, специфицируемыми на входе и выходе Д-оператора, понимаются множества данных и семейства таких множеств. Причем при спецификации нескольких множеств (семейств множеств) они записываются в виде $(D_1, ..., D_p)O(D'_1, ..., D'_s)$.

На основе модифицированной модели ЭВМ в работах [6–8] была предложена система алгоритмических алгебр (САА/Д), представляющая собой трехосновную алгебраическую систему $<U, L, D, \Omega>$, основами которой являются множество Д-операторов U, множество логических условий L и множество данных D, а $\Omega = \Omega_1 \cup \Omega_2 \cup \Omega_3$ – ее сигнатура, состоящая из Ω_1 – операций, принимающих значения на множестве U, Ω_2 – логических операций, принимающих значения на множестве L, Ω_3 – операций, принимающих значения на множестве данных D.

Из всего многообразия операций САА/Д, определенных на множестве Д-операторов, в данной статье воспользуемся следующими:

композиция Д-операторов $(D_i)O_i(D'_i)*$ $*(D_{i+1})O_{i+1}(D'_{i+1})$ (операция обозначается "*") означает последовательное выполнение сначала Д-оператора $(D_i)O_i(D'_i)$ и затем Д-оператора $(D_{i+1})O_{i+1}(D'_{i+1})$;

p-итерация $$\langle (D^{\pi})P(\alpha) \rangle \{(D)O(D')\} \qquad (1)$$

состоит в вычислении предиката $(D^{\pi})P(\alpha)$, где D^{π} – вектор заданного отношения, и проверке логического условия $\alpha \in \{1, 0\}$, продуцируемого предикатом. Если α истинно, выполняется Д-оператор $(D)O(D')$ и вновь вычисляется предикат и проверяется условие α. Циклический процесс, состоящий в проверке условия α и выполнении Д-оператора, осуществляется пока условие $\alpha = 1$, в противном случае операция *p*-итерации завершается.

В частном случае эта операция, записанная в виде

$$\langle 1 \rangle \{(D)O(D')\}, \qquad (2)$$

где 1 – тождественно истинное условие, реализует «бесконечный» цикл.

В работе [8] введен базовый набор данных – простые данные D_p, в частности указатели $a_{\tilde{D}}$, и составные данные: регулярно организованные данные – массивы одномерные (векторы) D_m и многомерные (*k*-мерные) D_{m^k}, записи

$D_з$, образованные с помощью простых данных, массивов и записей, массивы записей одномерные $D_{мз}$ и многомерные (k-мерные) $D_{мз^k}$. Элементы этого базового набора трактуются как множества данных.

Из множества операций, определенных в алгебре на множествах данных, в статье воспользуемся следующими.

Операция укрупнения данных (обозначается – *), определенная на произвольной совокупности произвольных множеств (семейств множеств) данных $\{D_1\},...,\{D_k\}$ (включая простые), образует новые данные $\{D_1\}*...*\{D_k\}=D$, представляющие собой семейство множеств $D=\{\{D_1\},...,\{D_k\}\}$.

Операция детализации данных ($\overline{*}$), определенная на произвольных семействах множеств (множествах) данных, за исключением простых, позволяет представить эти данные в виде совокупности множеств $\overline{*}D=\{D_1\},...,\{D_k\}$.

Отметим, что предложенные операции позволяют укрупнять (объединять) и детализовать (разбивать) данные, специфицированные на входе и выходе Д-операторов.

Представление любого исходного Д-оператора из U через образующие элементы системы $<U,L,D,\Omega>$ называется регулярной схемой этого Д-оператора (РСД). В частном случае, когда используется единственная операция – композиция, такое представление Д-оператора называется его композиционной схемой (КС).

Представление любых данных из D через образующие элементы системы $<U,L,D,\Omega>$ называется схемой данных (СД).

Реализации в рамках предложенного алгебраического аппарата комплексного подхода к разработке алгоритмов ИУС и посвящена данная статья.

Описание алгоритмов информационно-управляющих систем

Для самого общего случая определим алгоритм следующим образом.

Определение 2. Алгоритмом называется Д-оператор $(D)A(D')$, на входе которого специфицированы все необходимые для решения некоторой задачи глобальные данные, а на выходе – все результирующие глобальные данные.

При нисходящей стратегии проектирования [9] данные D и D' детализуются и записываются в виде СД, а алгоритм декомпозируется и, с учетом определения 2, записывается в виде РСД или КС. В последнем случае это выглядит как

$$(D)A(D')=$$
$$=(D_1)O_1(D_1')*(D_2)O_2(D_2')*...*(D_k)O_k(D_k'),$$

где $D=D_1\cup...\cup D_k$, $D'=D_1'\cup...\cup D_k'$, $(D_1)O_1(D_1')$, $...,(D_k)O_k(D_k')$ – производные Д-операторы, полученные в результате декомпозиции. На следующем шаге декомпозируются производные Д-операторы и детализуются специфицированные у них данные. Процесс продолжается до достижения требуемого уровня подробности описания алгоритма.

Традиционный (в первую очередь вычислительный) алгоритм представим в виде трех последовательно выполняемых подсистем (подзадач):

$$(D)A(D')=(D_п)O_п(D_п')*(D_т)O_т(D_т')*(D_з)O_з(D_з'),$$

где $O_п$ – выполняет подготовительные (стартовые) операции, $O_з$ – завершающие (заключительные) операции, $O_т$ – *тело* алгоритма. Такие алгоритмы назовем дискретными.

Алгоритм, характерный, в частности, для информационных систем, назовем непрерывно-дискретным и, используя (1), запишем в виде:

$$(D)A(D')=(D_п)O_п(D_п')*\left\langle(D^π)P(\alpha)\right\rangle\times$$
$$\times(D_т)O_т(D_т')*(D_з)O_з(D_з').$$

Циклическое выполнение тела алгоритма в данном случае прекращается, когда логическое условие α, продуцируемое предикатом $(D^π)P(\alpha)$, принимает ложное значение.

Известно, что управляющие системы (в частности, система логического управления [10]), в силу своей специфики, функционируют неограниченное время, т.е. их работа прекращается только в случае отключения питания. Алго-

ритмы такого рода назовем непрерывными и, используя (2), представим в виде:

$$(D)A(D') = (D_n)O_n(D'_n) * \langle 1 \rangle (D_m)O_m(D'_m), \quad (3)$$

где 1 – тождественно истинное логическое условие.

Исходя из рассмотренных типов алгоритмов, алгоритм ИУС определим следующим образом.

Определение 3. Алгоритм ИУС состоит из двух (или более) информационно связанных подсистем, которые реализуются в виде непрерывного алгоритма и выполняются параллельно (квазипараллельно) и асинхронно. Под информационной связью понимается двусторонняя связь, т.е. некоторое подмножество данных, продуцируемых одной подсистемой, используется второй и наоборот. По крайней мере, одна из подсистем взаимодействует с УВВ, а одна с УСО.

Основываясь на определении 3, вариант ИУС для случая, когда она состоит из двух подсистем, каждая из которых функционирует в собственной программно-аппаратной среде (двухмашинный случай) и реализуется непрерывным алгоритмом (3), архитектуру ИУС опишем в виде следующей РСД:

$$(D)\text{ИУС}(D') = ((D_{n\_\text{АСУ}})\Pi_{\text{АСУ}}(D'_{n\_\text{АСУ}}) *$$
$$*\langle 1 \rangle (D_{\text{АСУ}})\text{АСУ}(D'_{\text{АСУ}})) \dot{\vee} ((D_{n\_\text{САУ}}) \times$$
$$\times \Pi_{\text{САУ}}(D'_{n\_\text{САУ}}) * \langle 1 \rangle (D_{\text{САУ}})\text{САУ}(D'_{\text{САУ}})),$$

где $D = D_{n\_\text{АСУ}} \cup D_{\text{АСУ}} \cup D_{n\_\text{САУ}} \cup D_{\text{САУ}}$,

$D' = D'_{n\_\text{АСУ}} \cup D'_{\text{АСУ}} \cup D'_{n\_\text{САУ}} \cup D'_{\text{САУ}}$.

Реализацию подхода к проектированию алгоритма от данных рассмотрим на примере подсистемы АСУ, т.е. Д-оператора

$$(D_{\text{АСУ}})\text{АСУ}(D'_{\text{АСУ}}),$$

который преобразует входные данные $D_{\text{АСУ}}$ в выходные $D'_{\text{АСУ}}$. Для этого детализуем специфицированные на входе и выходе исходного Д-оператора данные, в соответствии с их структурой, записав их в виде СД: $\overline{*}D_{\text{АСУ}} = \{D_{\text{АСУ}1}\}, \ldots \quad \ldots, \{D_{\text{АСУ}n}\}$ и $\overline{*}D'_{\text{АСУ}} = \{D'_{\text{АСУ}1}\}, \ldots, \{D'_{\text{АСУ}n}\}$.

Далее, потребовав, чтобы их функциональность обеспечивала преобразование входных данных $D_{\text{АСУ}_p}$ в выходные $D'_{\text{АСУ}_s}$, введем производные Д-операторы вида $(D_{\text{АСУ}_p})\text{АСУ}_i(D'_{\text{АСУ}_s})$.

Для достижения требуемой функциональности, как правило, требуются вспомогательные (локальные) данные. Для того, чтобы учесть этот аспект, введем следующее определение.

Определение 4. Глобальными в КС $(D)O(D') = (D_1)O_1(D'_1) * \ldots * (D_k)O_k(D'_k)$ являются данные $D^G = D \cup D'$, специфицированные на входе и выходе исходного Д-оператора. На входе и выходе любого производного Д-оператора $(D_i)O_i(D'_i)$, в общем случае специфицируются как локальные, так и глобальные данные, т.е. $D_i = \{\{D_i^G\}, \{D_i^L\}\}$, $D'_i = \{\{D_i'^G\}, \{D_i'^L\}\}$, где $D_i^G, D_i'^G \subseteq D^G$. При этом локальные данные инкапсулированы в Д-операторе, т.е. невидимы и недоступны в других Д-операторах. Глобальные данные такие, что $(D_1^G \cup \ldots \cup D_k^G) = D$, $(D_1'^G \cup \ldots \cup D_k'^G) = D'$, а локальные такие, что $(D_1^L \cup \ldots \cup D_k^L) = D^L$, $(D_1'^L \cup \ldots D_k'^L) = D'^L$ и $D^L \bigcap \bigcap D^G = \varnothing$, $D'^L \bigcap D'^G = \varnothing$.

Исходя из определения 4, дополним специфицированные данные локальными ($D^L_{\text{АСУ}i}$, $D'^L_{\text{АСУ}i}$) в тех случаях, когда такие данные необходимы для выполнения требуемых преобразований.

Полученные в результате Д-операторы, в общем случае имеют вид:

$$(D_{\text{АСУ}_p}, D^L_{\text{АСУ}_i})\text{АСУ}_i(D'_{\text{АСУ}_s}, D^L_{\text{АСУ}i}), \quad (4)$$

а в частных –

$$(D_{\text{АСУ}_p})\text{АСУ}_i(D'_{\text{АСУ}_s}, D^L_{\text{АСУ}_i}),$$
$$(D_{\text{АСУ}_p}, D^L_{\text{АСУ}i})\text{АСУ}_i(D'_{\text{АСУ}_s}),$$
$$(D_{\text{АСУ}_p})\text{АСУ}_i(D'_{\text{АСУ}_s}). \quad (5)$$

Последовательность обработки данных в подсистеме задается следующей КС:

$$(D_{\text{АСУ}}, D^L_{\text{АСУ}})\text{АСУ}(D'_{\text{АСУ}}, D'^L_{\text{АСУ}}) =$$
$$= (D_{\text{АСУ}_s}, D^L_{\text{АСУ}1})\text{АСУ}_1(D'_{\text{АСУ}_p}, D'^L_{\text{АСУ}_1}) * \ldots *$$
$$*(D_{\text{АСУ}_m}, D^L_{\text{АСУ}n})\text{АСУ}_n(D'_{\text{АСУ}_r}, D^L_{\text{АСУ}n}),$$

где $\{D^L_{\text{АСУ}1}\}*...*\{D^L_{\text{АСУ}n}\} = D^L_{\text{АСУ}}$,

$\{D'^L_{\text{АСУ}_1}\}*...*\{D'^L_{\text{АСУ}_n}\} = D'^L_{\text{АСУ}}$.

На следующем шаге разработки специфицированные на входе и выходе у всех производных Д-операторов (4) локальные данные считаем как глобальные и Д-оператор, в связи с этим запишем в виде $({}^1D_{\text{АСУ}_i})^1\text{АСУ}_i({}^1D'_{\text{АСУ}_i})$. Последовательность действий повторяется аналогично описанному случаю, т.е. продолжается детализация данных. Например, в случаях ${}^1D_{\text{АСУ}_i}$ и ${}^1D'_{\text{АСУ}_i}$ получаем СД $\overline{*}{}^1D_{\text{АСУ}_i} = \{{}^2D_{\text{АСУ}_1}\},...$ $...,\{{}^2D_{\text{АСУ}_p}\}$ и $\overline{*}{}^1D'_{\text{АСУ}_i} = \{D'^2_{\text{АСУ}_1}\},...,\{D'^2_{\text{АСУ}_p}\}$, а для всех производных данных $\{{}^1D_{\text{АСУ}_s}\},...$ $...,\{{}^1D_{\text{АСУ}_m}\}$ и $\{{}^1D'_{\text{АСУ}_p}\}$, $...,\{{}^1D'_{\text{АСУ}_r}\}$ получаем семейство СД из $2n$ членов.

Специфицировав полученные глобальные данные на входах и выходах производных Д-операторов с соответствующей функциональностью и дополнив их локальными, получаем семейство КС из n элементов:

$$({}^1D_{\text{АСУ}_1})^1\text{АСУ}_1({}^1D'_{\text{АСУ}_1}) =$$
$$= ({}^2D_{\text{АСУ}_i}, {}^2D^L_{\text{АСУ}_1})^2\text{АСУ}_1({}^2D'_{\text{АСУ}_j}, {}^2D'^L_{\text{АСУ}_1})*...$$
$$...*({}^2D_{\text{АСУ}_s}, {}^2D^L_{\text{АСУ}_{p_1}})^2\text{АСУ}_{p_1}({}^2D'_{\text{АСУ}_r}, {}^2D'^L_{\text{АСУ}_{p_1}})$$

.....

$$({}^1D_{\text{АСУ}_n})^1\text{АСУ}_n({}^1D'_{\text{АСУ}_n}) =$$
$$= ({}^2D_{\text{АСУ}_g}, {}^2D^L_{\text{АСУ}\,p_{n-1}+1})^2\text{АСУ}_{p_{n-1}+1}({}^2D'_{\text{АСУ}_b}, {}^2D'^L_{ACY_{p_{n-1}+1}})*...$$
$$...*({}^2D_{\text{АСУ}_c}, {}^2D^L_{\text{АСУ}\,p_n})^2\text{АСУ}_{p_n}({}^2D'_{\text{АСУ}_m}, {}^2D'^L_{\text{АСУ}_{p_n}}),$$

где верхний левый индекс – номер шага (уровня) разработки алгоритма, а для производных Д-операторов выполнена сквозная нумерация (нижний правый индекс).

Такой процесс продолжается до достижения требуемого, например, *s*-го уровня детализации данных, после достижения которого получаем семейство КС, где *i*-я КС выглядит следующим образом:

$$({}^sD_{\text{АСУ}_i})^s\text{АСУ}_i({}^sD'_{\text{АСУ}_i}) =$$
$$= ({}^{s+1}D_{\text{АСУ}_p}, {}^{s+1}D^L_{\text{АСУ}_{m_{i-1}+1}})^{s+1}\text{АСУ}_{m_{i-1}+1}\times$$
$$\times({}^{s+1}D'_{\text{АСУ}_c}, {}^{s+1}D'^L_{\text{АСУ}_{m_{i-1}+1}})*....$$
$$...*({}^{s+1}D_{\text{АСУ}_r}, {}^{s+1}D^L_{\text{АСУ}_{m_i}})^{s+1}\times$$
$$\times\,\text{АСУ}_{m_i}({}^{s+1}D'_{\text{АСУ}_s}, {}^{s+1}D^L_{\text{АСУ}_{m_i}}).$$

В результате получаем описание алгоритма подсистемы в виде семейства КС, на всех этапах проектирования которого реализуется подход к разработке алгоритма от данных.

Функционально ориентированный подход (от управления) продемонстрируем на примере подсистемы САУ. Для этого разобьем задачу, решаемую подсистемой, на подзадачи, задавшись, таким образом, функциональностью производных Д-операторов, и определим последовательность их выполнения, записав КС, не специфицируя данные, в виде:

$$(D_{\text{САУ}})\text{САУ}(D'_{\text{САУ}}) = ()\text{САУ}_1()*...*()\text{САУ}_r().$$

После этого специфицируем для каждого производного Д-оператора в соответствии с его функциональностью, глобальные данные и, дополнив их в случае необходимости локальными, запишем КС в виде:

$$(D_{\text{САУ}})\text{САУ}(D'_{\text{САУ}}) =$$
$$= (D_{\text{САУ}_1}, D^L_{\text{САУ}_1})\text{САУ}_1(D'_{\text{САУ}_1}, D'^L_{\text{САУ}_1})*... \quad (6)$$
$$...*(D_{\text{САУ}_r}, D^L_{\text{САУ}_r})\text{САУ}_r(D'_{\text{САУ}_r}, D'^L_{\text{САУ}_r}).$$

Заметим при этом, что в частных случаях Д-операторы, образующие КС, могут выглядеть следующим образом:

$$(D_{\text{САУ}_i})\text{САУ}_i(D'_{\text{САУ}_i}, D'^L_{\text{САУ}_i}),$$
$$(D_{\text{САУ}_i}, D^L_{\text{САУ}_i})\text{САУ}_i(D'_{\text{САУ}_i}),$$
$$(D^L_{\text{САУ}_i})\text{САУ}_i(D'_{\text{САУ}_i}, D'^L_{\text{САУ}_i}),$$
$$(D_{\text{САУ}_i}, D^L_{\text{САУ}_i})\text{САУ}_i(D'^L_{\text{САУ}_i}), \quad (7)$$
$$(D_{\text{САУ}_i})\text{САУ}_i(D'_{\text{САУ}_i}),\ (D^L_{\text{САУ}_i})\text{САУ}_i(D'^L_{\text{САУ}_i}).$$

Потребуем, чтобы специфицированные глобальные данные $\{D_{\text{САУ}_1}\},...,\{D_{\text{САУ}_r}\}$ и $\{D'_{\text{САУ}_1}\},\ldots$ $...,\{D'_{\text{САУ}_r}\}$, в результате их укрупнения, образовывали исходные глобальные данные $\{D_{\text{САУ}}\}$ и $\{D'_{\text{САУ}}\}$.

Записав на первом шаге укрупнения эти данные в виде семейств СД:

$$\{D_{\text{САУ}_i}\}*...*\{D_{\text{САУ}_j}\} = {}^1D_{\text{САУ}_1}$$

.....

$$\{D_{\text{САУ}_c}\}*...*\{D_{\text{САУ}_m}\} = {}^1D_{\text{САУ}_k}$$

и

$$\{D'_{\text{САУ}_p}\}*...*\{D'_{\text{САУ}_s}\} = {}^1D_{\text{САУ}_1}$$

.....

$$\{D'_{\text{САУ}_u}\}*...*\{D'_{\text{САУ}_g}\} = {}^1D_{\text{САУ}_k}.$$

На втором – в виде семейства:

$$\{{}^2D_{\text{САУ}_p}\}*...*\{{}^2D_{\text{САУ}_c}\} = {}^2D_{\text{САУ}_1}$$

.....

$$\{{}^1D_{\text{САУ}_g}\}*...*\{{}^1D_{\text{САУ}_m}\} = {}^2D_{\text{САУ}_p},$$

и

$$\{{}^1D_{\text{САУ}_u}\}*...*\{{}^1D_{\text{САУ}_s}\} = {}^2D_{\text{САУ}_1}$$

.....

$$\{{}^1D_{\text{САУ}_k}\}*...*\{{}^1D_{\text{САУ}_m}\} = {}^2D_{\text{САУ}_p},$$

где, очевидно, $p < k$.

Будем продолжать этот процесс до достижения некоторого s-го шага, на котором получим соотношения:

$$\{{}^sD_{\text{САУ}_1}\}*...*\{{}^sD_{\text{САУ}_k}\} = D_{\text{САУ}}$$

и $$\{{}^sD'_{\text{САУ}_1}\}*...*\{{}^sD'_{\text{САУ}_r}\} = D'_{\text{САУ}}.$$

Что касается локальных данных, то для них выполняются соотношения:

$$\{D^L_{\text{САУ}1}\}*...*\{D^L_{\text{САУ}k}\} = D^L_{\text{САУ}},$$

$$\{D'_{\text{САУ}_1}\}*...*\{D'_{\text{САУ}_k}\} = D'^L_{\text{САУ}}.$$

КС (6), удовлетворяющая выдвинутому требованию, представляет собой первый шаг разработки алгоритма САУ, выполненный в рамках функционально ориентированного подхода.

Процесс разработки (декомпозиции) алгоритма повторяется на каждом шаге разработки аналогично рассмотренному случаю и продолжается до достижения требуемого, например, *p*-го уровня детализации, на котором будет получено семейство КС, где *i*-я КС выглядит следующим образом:

$$({}^pD_{\text{САУ}_i})\,{}^p\text{САУ}_i({}^pD'_{\text{САУ}_i}) =$$

$$= ({}^{p+1}D_{\text{САУ}_s}, {}^{p+1}D^L_{\text{САУ}_{n_{i-1}+1}})\,{}^{p+1}\text{САУ}_{n_{i-1}+1} \times$$

$$\times ({}^{p+1}D'_{\text{САУ}_c}, {}^{p+1}D'^L_{\text{САУ}_{n_{i-1}+1}})*...*({}^pD_{\text{САУ}_m}, {}^{p+1}D^L_{\text{САУ}_{n_i}})^{p+1} \times$$

$$\times \text{САУ}_{n_i}({}^{p+1}D'_{\text{САУ}_k}, {}^{p+1}D^L_{\text{САУ}_{n_i}}).$$

Полученное семейство КС представляет собой описание алгоритма подсистемы САУ, на всех этапах проектирования которого реализован функционально ориентированный подход.

Совместное использование двух подходов к разработке алгоритмов продемонстрируем на примере подсистемы $\Pi_{\text{САУ}}$, т.е. Д-оператор $(D_{n\_\text{САУ}})\Pi_{\text{САУ}}(D'_{n\_\text{САУ}})$.

Для этого детализируем специфицированные данные в соответствии с их структурой, записав их в виде СД:

$$\overline{*}D_{n\_\text{АСУ}} = \{D_{n\_\text{АСУ}1}\},...,\{D_{n\_\text{АСУ}p}\} \text{ и}$$

$$\overline{*}D'_{n\_\text{АСУ}} = \{D'_{n\_\text{АСУ}_1}\},...,\{D'_{n\_\text{АСУ}_r}\}.$$

Для некоторых подмножеств полученных данных $D_{n\_\text{АСУ}c},...,D_{n\_\text{АСУ}s}$, для которых очевидно, что в результате их преобразования будут продуцироваться данные $D'_{n\_\text{АСУ}_p},...,D'_{n\_\text{АСУ}_r}$, введем Д-операторы, аналогичные (4) и (5), функциональность которых такова, что они, с помощью введенных локальных данных, осуществляют такое преобразование, т.е. реализуем подход от данных.

Для остальных данных возможны, по крайней мере, два варианта развития событий.

Либо мы для каждой пары множеств данных $D_{n\_\text{АСУ}x}$ и $D'_{n\_\text{АСУ}_y}$ вводим некоторый промежуточный Д-оператор $(D_{n\_\text{САУ}_x})\tilde{\Pi}_{\text{САУ}}(D_{n\_\text{САУ}_y})$, либо укрупняем данные $\{D_{n\_\text{АСУ}g}\}*...*\{D_{n\_\text{АСУ}d}\} =$ $= D''_{n\_\text{АСУ}}$, $D'_{n\_\text{АСУ}_h}D'_{n\_\text{АСУ}_f} = D'''_{n\_\text{АСУ}}$ до получения двух множеств и вводим промежуточный Д-оператор $(D''_{n\_\text{САУ}})\Pi'_{CAУ}(D'''_{n\_\text{САУ}})$. К полученным таким образом Д-операторам применяем функционально ориентированный подход, аналогично рассмотренному случаю (6) – (7).

После этого задаем последовательность выполнения всех полученных Д-операторов, записав их в виде КС. Этот процесс, как и в описанных случаях, продолжается до достижения требуемого уровня детализации алгоритма.

В результате описание алгоритма подсистемы получаем как результат применения рассматриваемых подходов в сочетании.

Таким образом, показана возможность применения к разработке подсистем такого подхода, который наиболее полно отвечает ее специфике и соответствует традиционным взглядам на применение этих подходов.

Однако как было отмечено, затруднения часто возникают и в этом случае. Вместе с тем из приведенных рассуждений видно, что для любой подсистемы на любом из этапов разработки и для любой КС, полученной на этом этапе, можно делать выбор в пользу того подхода, который наиболее соответствует специфике каждой подзадачи, алгоритм которой описывается на данном этапе с помощью конкретной КС. Это, собственно, и обеспечивает комплексность подхода к разработке алгоритмов и есть основной результат данной статьи.

Заключение. Таким образом, определен класс алгоритмов ИУС и показана возможность реализации подхода от данных, функционально ориентированного подхода и их сочетания к разработке алгоритмов различных ее подсистем. Методы реализации этих подходов применимы на любом из этапов разработки, в любой последовательности и сочетании, при любом уровне детализации описания алгоритма. Такое их применение реализует комплексный подход к проектированию алгоритмов ИУС.

Внедрение комплексного подхода к разработке алгоритмов в практику может, по крайней мере, в некоторой степени положительно повлиять на положение дел в такой важной области человеческой деятельности как программирование.

Рассмотрение различных вариантов архитектуры конкретных ИУС, апробация комплексного подхода к их проектированию, шаги, направленные на внедрение предложенного подхода в практику, – направления дальнейших исследований.

1. *Дал У.*, *Дейкстра Э.*, *Хоор К.* Структурное программирование. – М.: Мир, 1975. – 247 с.
2. *Буч Г.* Объектно-ориентированный анализ и проектирование с примерами приложений на *C*++. – М.: Бином – СПб.: Невский диалект, 1998. – 560 с.
3. *Пьявченко Т.А.*, *Финаев В.И.* Автоматизированные информационно-управляющие системы. – Таганрог: Изд-во ТРТУ, 2007. – 271 с.
4. *Легалов А.И.* Процедурно-параметрическая парадигма программирования. Возможна ли альтернатива объектно-ориентированному стилю? – Красноярск, 2000. – 43 с. Деп. в ВИНИТИ 13.03.2000, № 622–В00.
5. *Коплиен Дж.* Мультипарадигменное проектирование для *C*++. Библиотека программиста. – СПб.: Питер, 2005. – 235 с.
6. *Акуловский В.Г.* Основы алгебры алгоритмов, базирующейся на данных // Проблеми програмування. – 2010. – № 2–3. – С. 89– 96.
7. *Акуловский В.Г.* Алгебра алгоритмов, базирующаяся на данных // Кибернетика и системный анализ. – 2012. – № 2. – С. 151–166.
8. *Акуловский В.Г.* Алгебра для описания данных в композиционных схемах алгоритмов // Поблеми програмування. – 2012. – № 2–3. – С. 234 – 240.
9. *Дорошенко А.Е.*, *Акуловский В.Г.* Нисходящее проектирование алгоритмов в рамках алгероалгоритмического подхода // Математические машины и системы. – 2012. – № 3. – С. 97–102.
10. *Закревский А.Д.* Параллельные алгоритмы логического управления. – М. : Эдиториал УРСС, 2012. – 200 с.

Поступила 29.04.2013

Тел. для справок: +38 044 526-3559,
+38 050 941-0566 (Киев)
E-mail: *valeryakulovskiy@rambler.ru*