

Н.П. Вашкевич, В.Н. Дубинин

ВОПРОСЫ РАЗРАБОТКИ ОПЕРАЦИОННОЙ СЕМАНТИКИ ФУНКЦИОНАЛЬНЫХ БЛОКОВ IEC 61499¹

Аннотация. В работе идентифицируется проблема определения формальной семантики функциональных блоков (ФБ) стандарта IEC 61499 и разрабатываются концептуальные основы для построения модели операционной семантики функциональных блоков на базе машин абстрактных состояний. Дается формальное определение нотации для представления операционной семантики ФБ, а также рассматриваются варианты функционально-структурной организации семантических моделей систем ФБ. Предложенная нотация может быть использована для определения операционной семантики систем ФБ, функционирующих в рамках циклической, последовательной, синхронной, а также других моделей выполнения.

Ключевые слова: функциональный блок, стандарт IEC 61499, операционная семантика, машины абстрактных состояний.

Введение

Существующее положение дел в сфере семантики нового языка программирования распределенных контроллеров IEC 61499 [5] во многом определяется следующими факторами: 1) наличием пробелов в определении работы функциональных блоков (ФБ) в тексте самого стандарта IEC 61499; 2) наличием большого числа моделей выполнения ФБ; 3) отсутствием общепризнанной формальной семантики ФБ. Отсутствие целостного определения формальной семантики ФБ порождает целый ряд проблем: 1) непонимание детальной работы ФБ приводит к различным интерпретациям ФБ и различным реализациям программно-аппаратных платформ, что в свою очередь может привести к проблеме портбельности управляющего программного обеспечения (ПО); 2) проведение верификации

и имитационного моделирования на основе упрощенных (и, возможно, ошибочных) моделей может привести к неправильным результатам, что, в свою очередь, напрямую отражается на надежности управляющего ПО; 3) рост скептицизма по отношению к стандарту IEC 61499 и замедление темпов его внедрения в промышленную практику.

Сложность ФБ в семантическом плане определяется следующими факторами: 1) наличием различных, порой нетривиальных, моделей выполнения; 2) наличием алгоритмов обработки данных в базисных ФБ; 3) выполнением ФБ на основе управления событиями (*event driven execution*); 4) иерархичностью и модульностью системы ФБ; 5) наличием типизации и инстанциации большинства артефактов проектирования (ФБ, субприложений, ресурсов, устройств); 6) наличием у ФБ свойств как программного модуля, так и аппаратного; 7) тем, что язык ФБ является *визуальным* языком программирования, использующим блок-диаграммы с встроенной машиной состояний, неограни-

¹ Работа выполнена в рамках федеральной целевой программы «Научные и научно-педагогические кадры инновационной России» на 2009-2013 годы».

ченным иерархическим вложением блоков друг в друга и их распределением по сетевым устройствам, использованием принципа потока событий как основы вычислений. Подобные визуальные компонентно-базированные событийно-ориентированные языки требуют особых подходов к описанию их семантики.

Существующие модели ФБ на основе *NCES*-сетей, конечных и временных автоматов, сетей Петри, нельзя отнести к формальной семантике ввиду их абстрактности. В работе [2] семантика ФБ представлена в виде системы переходов состояний, но не в полной мере (без учета конкретных моделей выполнения). В работе [1] предложена формальная рамочная нотация для моделирования ФБ в различных моделях выполнения. В принципе данную модель можно считать формальной семантической моделью, но с некоторыми оговорками: 1) не рассматривается *OSM*-машина, играющая ключевую роль в выполнении базисного ФБ и определенная в стандарте IEC 61499; 2) предложенная модель является довольно абстрактной в ряде позиций.

Существует множество подходов к определению формальной семантики, но основными являются следующие классы формальных семантик: денотационная, операционная и аксиоматическая [6]. Наибольший интерес в плане реализации представляет *операционная семантика*. В последнее время все большее распространение для определения операционной семантики языков находит аппарат машин абстрактных состояний (МАС), введенный Ю.Гуревичем [4]. Эта формальная нотация имеет ряд преимуществ перед другими методами: 1) выразительность, простота и понятность; 2) возможность представления алгоритмов на различных уровнях; 3) формальное описание может использоваться как *выполнимая спецификация*, что обеспечивает легкость создания различных сред выполнения. К настоящему времени предложено множество вариантов

МАС, среди которых недетерминированные, параллельные, синхронные, асинхронные, мультиагентные, распределенные и интерактивные МАС [4]. МАС были успешно использованы при определении формальной семантики различных языков программирования и моделирования, а также непроцедурных языков спецификаций (например, языка SDL [3]).

Таким образом, на основе изложенного выше анализа выбираем подход к определению формальной семантики ФБ на основе модифицированных распределенных МАС. Данный подход развивается авторами в [7,8,9]. Ниже излагаются концептуальные вопросы, не нашедшие отражения в этих работах, связанные: 1) с формальной моделью (нотацией) для определения операционной семантики ФБ (ФМОСФБ) на основе МАС; 2) функционально-структурной организацией моделей операционной семантики ФБ.

1 Модульная формальная модель операционной семантики ФБ

Можно выделить следующие особенности ФМОСФБ: 1) использование переменных состояния и функций значений переменных состояния при определении состояния системы; 2) асинхронность модели; 3) мультиагентность, причем число (неявных) агентов равно числу модулей; 4) наличие общих переменных у модулей; 5) детерминированность (функциональность) модулей; 6) использование явных продукционных правил при представлении программы МАС; 7) особое ограничение на выполнение распределенной МАС.

При определении ФМОСФБ будут использоваться следующие соглашения об обозначениях. Пусть $Z_A: A \rightarrow Dom(A)$ – функция, назначающая объектам из множества A значения (из области допустимых значений $Dom(A)$). Тогда $[Z_A]$ будем обозначать множество всех возможных функций Z_A' . Число возможных функций из $[Z_A]$ определяется как число раз-

мещений с повторениями из n по k и равно n^k , где $k=|A|$, $n=|Dom(A)|$. Наличие взаимно однозначного соответствия между множествами A и B показывается как $A \Leftrightarrow B$. Придерживаясь концепции и нотации машин абстрактных состояний, введем оператор обновления функции значений, обозначив его \leftarrow . Данный оператор может быть определен следующим образом: $Z_A(a) \leftarrow b \triangleq (Z_A \setminus (a, x)) \cup (a, b)$, где $a \in A$; $b, x \in Dom(A)$; $Z_A \subseteq A \times Dom(A)$ – график функции. Здесь символ \triangleq читается, как «есть по определению».

По структуре ФМОСФБ представляет совокупность асинхронно работающих (синхронных) модулей:

$$W = (M^1, M^2, \dots, M^n)$$

Каждый модуль $M^j \in W$ определяется следующим кортежем (во внутренних объектах верхний индекс j для простоты опущен):

$$M^j = (V, (Dom(v_i))_{v_i \in V}, (T_{v_i})_{v_i \in V}, (p_{v_i})_{v_i \in V}, (Z_{v_i}^0)_{v_i \in V})$$

где $V = \{v_1, v_2, \dots, v_m\}$ – множество переменных состояния модуля;

$Dom(v_i)$ – область допустимых значений переменной $v_i \in V$;

T_{v_i} – функция для вычисления значений переменной состояния $v_i \in V$. Данная функция может быть представлена в глобальном или локальном варианте. В первом случае функция определяется как: $T_{v_i}^G : \prod_{v_k \in V} [Z_{v_k}] \rightarrow Dom(v_i)$. Поскольку не все переменные влияют на изменение какой-либо другой переменной, то глобальная функция является избыточной и может быть сведена к локальному варианту: $T_{v_i}^L : \prod_{v_k \in H(v_i)} [Z_{v_k}] \rightarrow Dom(v_i)$,

где $H(v_i) = \{v_{i_1}, v_{i_2}, \dots, v_{i_q}\}$ – множество переменных $v_{i_j} \in V$, $j=1, q$, которые влияют на изменение переменной $v_i \in V$;

$p_{v_i} \triangleq Z_{v_i}(v_i) \leftarrow T_{v_i}(Z_{v_{i_1}}, Z_{v_{i_2}}, \dots, Z_{v_{i_q}})$ – правило обновления функции значений переменной $v_i \in V$;

$Z_{v_i}^0$ – функция начального значения переменной $v_i \in V$.

Выполнение модуля M^j заключается в одновременном (синхронном) выполнении всех правил $(p_{v_i})_{v_i \in V}$.

Состояние S системы определяется переменными, входящими во все модули W , а также их значениями: $S = \prod_{v_k \in V^S} Z_{v_k}$, где $V^S = \bigcup_{i=1}^n V^i$ –

множество переменных системы. Как известно, в распределенных МАС порядок выполнения модулей не определяется, только задаются ограничения на этот порядок [5]. Порядок выполнения модулей ФМОСФБ может быть произвольным, но с единственным ограничением – в прогоне ФМОСФБ не должно быть выполнения модулей, не изменяющих текущего состояния. Для обеспечения этого требования предлагаются следующие реализационные схемы (стратегии) выполнения модулей:

1) выполняется тот модуль, у которого изменились входные данные, которые, в свою очередь, могут повлиять на изменение состояния. Из множества переменных V^i модуля M^i выделим общие переменные с другими модулями: $V_{COM}^i = V^i \cap \bigcup_{j \neq i} V^j$. В свою очередь из этих переменных выделим: $V_{RD}^i \subseteq V_{COM}^i$ – множество переменных, которые влияют на изменение текущего состояния при выполнении модуля M^i ; $V_{WR}^i \subseteq V_{COM}^i$ – множество переменных, значения которых изменяет модуль M^i при своем выполнении. Следует отметить, что общие переменные модулей возникают как следствие наличия вертикальных и горизонтальных связей в моделируемой иерархической системе ФБ. Обозначим $Z_{RD_{curr}}^i$ – упорядоченное множество текущих значений переменных из V_{RD}^i ; $Z_{RD_{old}}^i$ – упорядоченное множество значений переменных из V_{RD}^i при предыдущем прогоне. Тогда условие запуска модуля ФМОСФБ формально определяется как $Z_{RD_{curr}}^i \neq Z_{RD_{old}}^i$.

2) после выполнения модуля M^i могут быть выполнены модули, на которые модуль M^i оказывает непосредственное влияние по

переменным, то есть модули из множества: $M_{SUCC} = \{M^j \mid V_{WR}^i \cap V_{RD}^j \neq \emptyset, j = \overline{1, n}\}$. Это следствие принципа локальности изменения глобального состояния при выполнении модуля.

Очевидно, что модули ФМОСФБ должны быть «робастными» в смысле нечувствительности к порядку запуска других модулей. С учетом робастности модуля можно предположить «транзакционный» принцип выполнения модуля ФМОСФБ, согласно которому модуль выполняется не однократно (как в МАС), а возможно многократно – до достижения неподвижной точки (*fixpoint*), то есть до тех пор, пока повторные выполнения модуля не будут приводить к изменениям переменных состояния, локализованных в модуле. Использование этого принципа не повлияет на конечный результат, но может быть удобно при программной реализации системы.

ФМОСФБ может быть использована для описания семантики систем ФБ, функционирующих в соответствии с различными моделями выполнения. Правила изменения функций значений переменных, в ФМОСФБ в самом общем виде, будут детализироваться с использованием продукционных правил. В самом простом случае одному правилу изменения функции будет соответствовать одно продукционное правило. Общий вид продукционного правила: $p_i^m : c \Rightarrow a$, где p_i^m – идентификатор правила (или группы правил); c – условие применения правила, a – действия по изменению переменной. Идентификатор правила включает: t – имя изменяемой переменной; m – модификатор идентификатора правила, позволяющий получить дополнительную идентифицирующую информацию о правиле. Модификатор представляется в формате B, C, D или B, C , где B – номер правила среди правил по изменению переменной t ; C – идентификатор модуля, в котором локализовано правило; D – идентификатор используемой модели выполнения (для модулей диспетчера). Условия в левых

частях правил по изменению одной и той же переменной являются *взаимоисключающими*, поэтому коллизий правил по записи не происходит. Порядок записи правил является несущественным. Для краткости представления однотипных правил может использоваться параметризация правил и группирование параметризованного правила в виде множества.

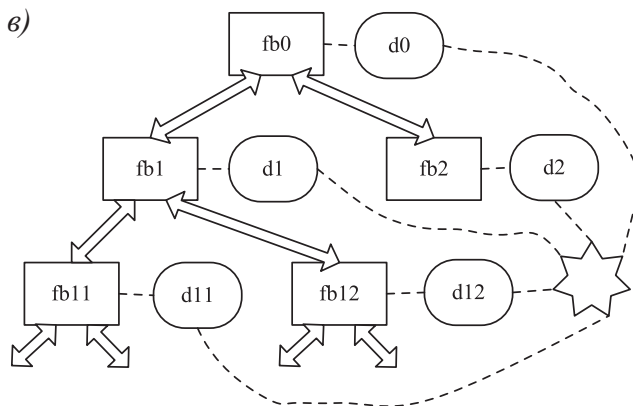
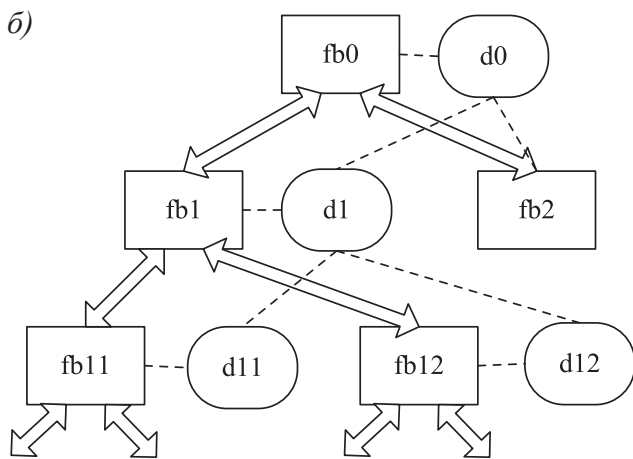
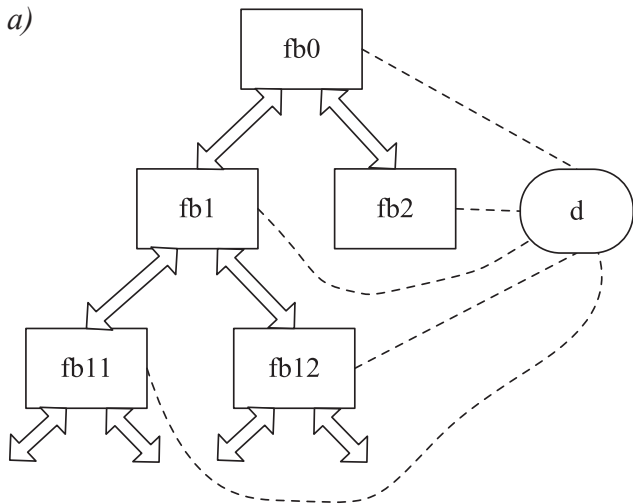
Для удобства могут использоваться группы приоритетных правил для изменения переменной. Группа правил с приоритетами заключается в угловые скобки: $\langle p_1 : c_1 \Leftrightarrow a_1; p_2 : c_2 \Leftrightarrow a_2; \dots p_n : c_n \Leftrightarrow a_n \rangle$. Следует заметить, что приоритетная группа правил легко может быть преобразована в “бесприоритетную” группу: $\langle p_1 : c_1 \Rightarrow a_1; p_2 : c_2 \wedge \overline{c_1} \Rightarrow a_2; \dots p_n : c_n \wedge \overline{c_1} \wedge \overline{c_2} \wedge \dots \wedge \overline{c_{n-1}} \Rightarrow a_n \rangle$

2 Функционально-структурная организация моделей операционной семантики систем ФБ

При определении семантики ФБ следует учитывать две составляющие: 1) работу собственно ФБ, неформально описанную в стандарте IEC 61499; 2) работу системы управления выполнением ФБ на ресурсе, обеспечивающую заданную модель выполнения ФБ. Перед разработкой семантики выполнения ФБ следует оценить в целом возможную функционально-структурную организацию формальной модели с целью: 1) выделить инвариантные и изменяемые модули с тем, чтобы облегчить описание совокупности семантик ФБ для различных моделей выполнения путем повторного использования инвариантных описаний; 2) использовать полученную функционально-структурную организацию при реализации инструментальных сред выполнения (run-time).

На рис. 1 на примере системы ФБ приведены возможные варианты функционально-структурной организации семантических моделей систем ФБ. На данном рисунке под fb0, fb1, fb11 и fb12 понимаются операционные модели составных ФБ или субприложений, а

под fb2 – операционная модель базисного ФБ. Широкие дуги представляют естественные потоки событий и данных между ФБ смежных уровней, а пунктирные линии – информацию, используемую в управлении выполнением ФБ.



С точки зрения структурной организации самым простым является вариант с общим диспетчером (вариант а), однако это, как правило, наиболее сложный вариант по функциональности. В варианте б) диспетчирование представлено совокупностью иерархически взаимосвязанных диспетчеров. Для некоторых моделей выполнения вариант б) допускает простой переход от одной модели выполнения к другой путем замены диспетчера. Вариант в) определяет систему слабосвязанных диспетчеров, взаимодействующих через общую память.

Заключение

В работе дано формальное определение нотации для представления операционной семантики ФБ стандарта IEC 61499, основанной на машинах абстрактных состояний, а также определены варианты функционально-структурной организации семантических моделей систем ФБ. Следует отметить, что отмеченная нотация была апробирована при определении операционной семантики систем ФБ, функционирующих в рамках последовательной [7] и синхронной двухтактной [8,9] моделей выполнения.

Рис. 1. Функционально-структурная организация моделей систем ФБ:
 а – с общим диспетчером;
 б – с иерархически связанными диспетчерами;
 в – с независимыми диспетчерами, взаимодействующими через общую память

Библиография:

1. Cengic, G. On Formal Analysis of IEC 61499 Applications, Part A: Modeling / G. Cengic, K. Åkesson // IEEE Transactions on Industrial Informatics. – 2010. – Vol. 6, No. 2. – P. 136-144.
2. Dubinin, V. On Definition of a Formal Semantic Model for IEC 61499 Function Blocks / V. Dubinin, V. Vyatkin // EURASIP Journal on Embedded Systems. – 2008. – Vol. 2008. – 10 p.
3. Glässer, U. Abstract State Machine Semantics of SDL / U. Glässer, R. Karges // Journal of Universal Computer Science. – 1997. – Vol. 3, No. 12. – P. 1382-1414.
4. Gurevich, Y. Evolving Algebras 1993: Lipari Guide / Y. Gurevich // Specification and Validation Methods. – Oxford University Press, 1995. – P.9-36.
5. International Standard IEC 61499. Function blocks for industrial-process measurement and control systems. Part 1: Architecture / International Electrotechnical Commission. – Geneva, 2005. – 111 p.
6. Mosses, P. D. Formal Semantics of Programming Languages: An Overview / P. D. Mosses // Electronic Notes in Theoretical Computer Science. – 2006. – Vol. 148, Issue 1. – P. 41–73.
7. Вашкевич, Н.П. Формализованное описание последовательной модели выполнения функциональных блоков / Н.П. Вашкевич, В.Н. Дубинин, В.В. Вяткин // Межвуз. сб. науч. тр. «Вычислительные системы и технологии обработки информации». – 2011. – Выпуск 10. – С. 45-61.
8. Дубинин, В.Н. Операционная семантика синхронных функциональных блоков IEC 61499 на основе машин абстрактных состояний, Часть 1: Модель диспетчеров / В.Н. Дубинин, В.В. Вяткин // XXI век: итоги прошлого и проблемы настоящего Плюс. – 2012. – Вып. 4. – С. 233-240.
9. Дубинин, В.Н. Операционная семантика синхронных функциональных блоков IEC 61499 на основе машин абстрактных состояний, Часть 2: Модели блоков и представление в SMV / В.Н. Дубинин, В.В. Вяткин // Труды Международной науч.-техн. конф. “Современные информационные технологии”. – Пенза, 2011. – Вып.14 – С. 94-100.

References (transliteration):

1. Cengic, G. On Formal Analysis of IEC 61499 Applications, Part A: Modeling / G. Cengic, K. Åkesson // IEEE Transactions on Industrial Informatics. – 2010. – Vol. 6, No. 2. – P. 136-144.
2. Dubinin, V. On Definition of a Formal Semantic Model for IEC 61499 Function Blocks / V. Dubinin, V. Vyatkin // EURASIP Journal on Embedded Systems. – 2008. – Vol. 2008. – 10 p.
3. Glässer, U. Abstract State Machine Semantics of SDL / U. Glässer, R. Karges // Journal of Universal Computer Science. – 1997. – Vol. 3, No. 12. – P. 1382-1414.
4. Gurevich, Y. Evolving Algebras 1993: Lipari Guide / Y. Gurevich // Specification and Validation Methods. – Oxford University Press, 1995. – P.9-36.
5. International Standard IEC 61499. Function blocks for industrial-process measurement and control systems. Part 1: Architecture / International Electrotechnical Commission. – Geneva, 2005. – 111 p.
6. Mosses, P. D. Formal Semantics of Programming Languages: An Overview / P. D. Mosses // Electronic Notes in Theoretical Computer Science. – 2006. – Vol. 148, Issue 1. – P. 41–73.

7. Vashkevich, N.P. Formalizovannoe opisanie posledovatel'noy modeli vypolneniya funktsional'nykh blokov / N.P. Vashkevich, V.N. Dubinin, V.V. Vyatkin // Mezhvuz. sb. nauch. tr. «Vychislitel'nye sistemy i tekhnologii obrabotki informatsii». –2011. – Vypusk 10. – S. 45-61.
8. Dubinin, V.N. Operatsionnaya semantika sinkhronnykh funktsional'nykh blokov IEC 61499 na osnove mashin abstraktnykh sostoyaniy, Chast' 1: Model' dispatcherov / V.N. Dubinin, V.V. Vyatkin // XXI vek: itogi proshlogo i problemy nastoyashchego Plyus. – 2012. – Vyp. 4. – S. 233-240.
9. Dubinin, V.N. Operatsionnaya semantika sinkhronnykh funktsional'nykh blokov IEC 61499 na osnove mashin abstraktnykh sostoyaniy, Chast' 2: Modeli blokov i predstavlenie v SMV / V.N. Dubinin, V.V. Vyatkin // Trudy Mezhdunarodnoy nauch.-tekhn. konf. “Sovremennye informatsionnye tekhnologii”. – Penza, 2011. – Vyp.14 – S. 94-100.