

УДК 004.415.2

## АНАЛИЗ ОРГАНИЗАЦИИ ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА В СТАНДАРТЕ IEC 61499

А.Н. Лукичев

В статье проводится анализ организации вычислительного процесса в распределенных системах управления, описываемой стандартом IEC 61499. Выявляются основные недостатки организации и описываются способы их решения.

**Ключевые слова:** распределенные системы управления, модели вычислений, встроенные системы.

### Введение

Стандарт IEC 61499 [1] был принят в 2005 г. Международной электротехнической комиссией после почти 10 лет предварительного обсуждения. Стандарт описывает организацию распределенных систем мониторинга и управления промышленными процессами, предоставляет языковые средства для их описания и требования к реализации на вычислительной платформе. В настоящее время организационной и исследовательской деятельностью по развитию стандарта занимается некоммерческий консорциум O<sup>3</sup>NEIDA при поддержке IEC. Различные вопросы, связанные с использованием IEC 61499, регулярно обсуждаются на таких конференциях, проводимых IEEE, как ETFA (Emerging Technologies and Factory Automation), ISORC (International Consortium on Ob-

ject-oriented Real-time distributed Computing), INDIN (Conference on Industrial Informatics) и другие.

Несмотря на значительное время, в течение которого стандарт обсуждался перед принятием, множество вопросов организации вычислений в распределенных системах управления (PCY) осталось за рамками стандарта. Многие из них препятствуют использованию в полной мере таких основных достоинств IEC 61499 по сравнению с широко используемым в настоящее время IEC 61131, как распределенность и параллелизм вычислений. В статье проводится краткий анализ и подчеркиваются существенные проблемы организации вычислительного процесса, регламентируемой стандартом.

### Функциональные блоки

Функциональный блок (ФБ) представляет собой компонент вычислительного процесса, воспринимающий входные сигналы, активизирующий вычисления и генерирующий выходные сигналы в зависимости от наличия событий на входах и своего состояния. У ФБ должен быть хотя бы один вход, воспринимающий события, по которым он активизирует вычисления. У ФБ может быть один или несколько входов данных, один или несколько выходов событий, генерируемых по завершении вычислений, и один или несколько выходов данных, состояние которых может обновляться также по завершении вычислений.

Стандарт описывает ФБ трех типов: простые (basic), сервисные (service) и составные (composite). Простой ФБ содержит набор алгоритмов обработки данных, внутренние данные и логику их выбора и активизации, описываемую управляющим конечным автоматом (ЕСС, Execution Control Chart). Основные элементы простого ФБ изображены на рис. 1. Алгоритмы простого ФБ могут использовать входы данных и внутренние данные, генерируя на выходах результаты вычислений. Переходы между состояниями ЕСС могут быть активизированы по событиям на входах событий. С каждым состоянием ЕСС может быть связано одно или несколько действий, выполняемых при переходе в него. Каждое такое действие представляет собой активизацию одного из алгоритмов и/или генерацию события на одном или нескольких выходах событий по завершении алгоритма.

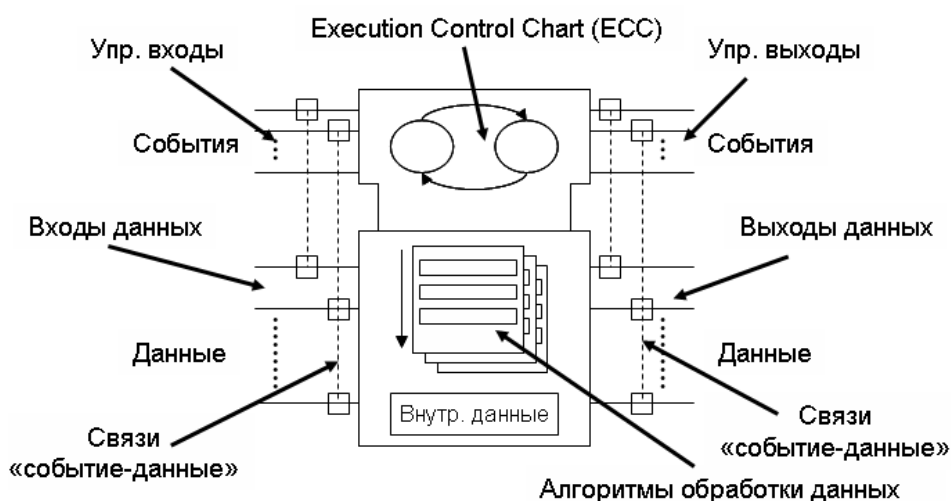


Рис. 1. Простой (basic) функциональный блок

Для обеспечения контроля за взаимодействием ФБ в распределенной системе стандарт требует, чтобы каждый вход данных был связан с входом событий, если по

событию на нем может быть выбран алгоритм, который будет использовать указанный вход данных. Стандарт требует, чтобы все входы данных были связаны хотя бы с одним из входов событий. То же самое касается и выходов данных. Если после обновления данных на выходе может быть сгенерировано событие на выходе событий, то указанный выход данных должен быть связан с выходом событий, и все выходы данных ФБ должны быть связаны хотя бы с одним выходом событий.

Стандарт также предусматривает существование сервисных ФБ, предназначенных для отображения коммуникационных и других сервисов платформы на функциональные блоки. При этом регламентируется лишь внешний интерфейс сервисного ФБ (всего для двух типов – для коммуникационных и для управления приложением, хотя платформа может предоставлять и другие, совершенно различные сервисы), но его внутренняя реализация не регламентируется. Для приложения сервисный ФБ выглядит как простой ФБ, в котором отсутствует спецификация ЕСС и алгоритмов.

Составной ФБ отличается от простого тем, что вместо набора алгоритмов и ЕСС он содержит сеть простых, сервисных и составных ФБ. Важно отметить, что для составного ФБ также необходимо указывать связи между входами данных и событий и выходами данных и событий. При этом они не обязательно должны соответствовать связям в его внутренней сети.

### Поведение ФБ

Несмотря на то, что проект стандарта обсуждался почти 10 лет перед тем, как был принят в 2005 г., многие вопросы организации вычислений в описываемой им распределенной системе остались открытыми. Исследователи отмечают возможность неоднозначного понимания поведения отдельных ФБ и координации вычислений, приводящую к несовместимости приложений, платформ и инструментов проектирования от различных производителей [2, 4, 5].

Рассмотрим управляющий автомат простого ФБ на рис. 2. Пусть в начальный момент времени автомат находится в состоянии State1, а условие Cond, формируемое внутренними данными ФБ, выполняется.

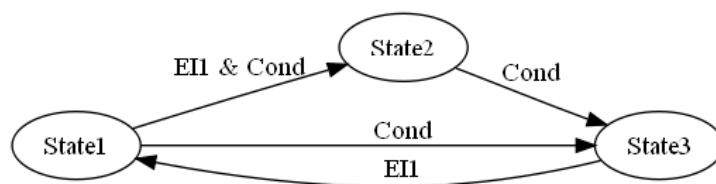


Рис. 2. Пример ЕСС, допускающего неоднозначное поведение в рамках IEC 61499

Стандарт требует, чтобы анализ выполнения условий переходов ЕСС производился после возникновения одного или нескольких событий на входе ФБ. Таким образом, автомат может перейти в другое состояние только при возникновении события на одном из входов событий, например на входе E1. Однако в этом случае будет возможен переход как в State2, так и в State3. Согласно стандарту, анализ условий переходов выполняется в любом заранее заданном порядке, например, в порядке перечисления переходов в формальной спецификации ЕСС. Это вносит неоднозначность в спецификации ЕСС с помощью диаграмм, так как часто при создании диаграмм переходов между состояниями руководствуются принципами читаемости и компактности, не имеющими ничего общего с поведением системы. Эта проблема может возникнуть и при автоматизированном создании графического представления ЕСС из текстовых описаний. Тем не

менее, пусть в рассматриваемый момент времени выбирается переход в состояние State3, и событие на входе EI1 остается «непрочитанным». Стандарт не отвечает на вопрос, будет ли сразу после этого возможен переход обратно в состояние State1 или «непрочитанные» события теряются при переходе между состояниями.

С другой стороны, допустим, что был выбран переход в состояние State2, и после его выполнения значение условия Cond стало равно false. Из состояния State2 возможен единственный переход по условию Cond, которое зависит только от внутренних данных ФБ и может изменить свое значение только после очередного перехода в результате выполнения соответствующего алгоритма. В данном случае изменение Cond невозможно, и ЕСС останется в состоянии State2 при возникновении любых событий на входе ФБ. Поскольку выбор алгоритма и его активизация производится только один раз при переходе ЕСС в одно из состояний, в данном случае при возникновении любых событий на входе ФБ, когда его ЕСС находится в состоянии State2, не будет производиться обработка данных. ФБ будет заблокирован. Несмотря на тривиальность приведенного примера, подобные ситуации могут возникнуть в неочевидных случаях и вполне допускаются стандартом [2].

### Сеть ФБ

Стандарт не решает множество вопросов координации вычислений в сети ФБ. К примеру, стандарт допускает существование сети, изображенной на рис. 3. Как видно из рисунка, обновление данных на входе двух ФБ-приемников может быть не согласовано с событиями на их входе. Это может привести к нестабильности исходных данных вычислений, производимых ФБ. В процессе выполнения алгоритма в ФБ-приемнике данные на его входе могут измениться без появления соответствующего события, так как оно генерируется не источником данных.

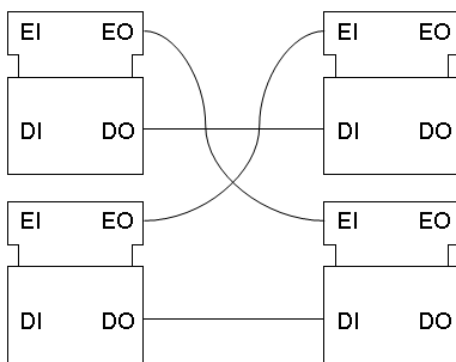


Рис. 3. Сеть ФБ, в которой данные не согласованы с событиями

Кроме чрезмерной универсальности предполагаемой стандартом модели вычислений, эта модель представляется довольно усложненной. С одной стороны, для простых ФБ внутренняя реализация довольно подробно описывается, а с другой, для сервисных ФБ она совершенно не регламентируется. Кроме того, составной ФБ в иерархической модели не может быть представлен как простой ФБ, так как простой ФБ должен иметь текущее состояние управляющего автомата и набор алгоритмов. Очевидно, далеко не для любой сети ФБ, реализующей составной ФБ, можно задать эти два компонента (тем более что в ней могут присутствовать сервисные ФБ).

Это обстоятельство существенно затрудняет анализ, верификацию и реализацию на платформе вычислительного процесса, описываемого в рамках стандарта, и вынуждает прибегать к преобразованию иерархической модели (в которой присутствуют со-

ставные ФБ) в «плоскую» (в которой они отсутствуют) [5, 6]. При таком преобразовании возникает сложность перехода от составного ФБ к реализующей его сети. Как видно из рис. 4, наличие у составного ФБ собственных связей между событиями и данными не позволяет в общем случае просто заменить его на его внутреннюю сеть. Требуется создание специальных ФБ-адаптеров, обеспечивающих такие связи.

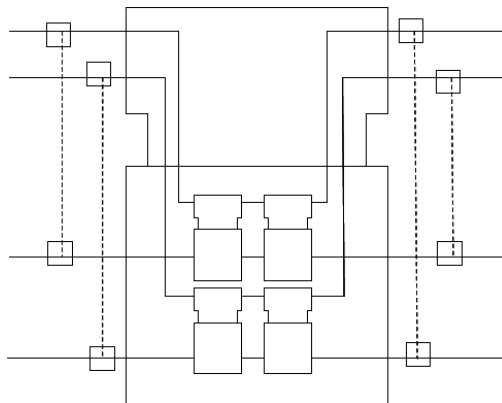


Рис. 4. Составной ФБ и его связи «событие-данные»

### Требования реального времени

Стандарт называет такие временные характеристики простых ФБ, как WCET (Worst-Case Execution Time, максимальное время выполнения) алгоритмов, время реакции ЕСС на событие, минимальные интервалы между входными событиями. Тем не менее, этим характеристикам не уделяется достаточного внимания. В стандарте нет ни ограничений, ни описания зависимостей между ними, ни правил, которым эти и другие временные характеристики должны удовлетворять. Таким образом, временной аспект поведения системы остается за рамками стандарта.

Это однозначно неприемлемо при проектировании управляющих систем, которые в большинстве являются системами реального времени. Не имея возможности судить о временных характеристиках отдельного ФБ, невозможно говорить о времени ответа, задержке управления и других важных свойствах проектируемой управляющей системы. Поскольку временные свойства системы не регламентируются стандартом, они автоматически становятся зависимыми от инструментов автоматизированного синтеза и используемой вычислительной платформы. Это существенно снижает портируемость проектов, несмотря на то, что это свойство декларируется как одно из основных достоинств стандарта.

### Координация распределенных вычислений

Другим основным достоинством стандарта является акцент на распределенность вычислительного процесса, однако и этим вопросам в нем уделено недостаточно внимания. Речь идет не об общей организации системы (сеть устройств, каждое устройство содержит ресурсы, каждый ресурс выполняет вычисления, а приложение отображается на эту совокупность ресурсов и т.д.), а о вопросах координации распределенных вычислений. Декларируется, что потенциально коммуникационный сервис в системе не может считаться надежным (события и данные в распределенной системе могут теряться или доставляться с непредсказуемыми задержками), однако не оговариваются действия при потере или превышении допустимого времени доставки информации. Не предлагается никаких способов решения этой проблемы.

Несмотря на декларируемую простоту распределения компонентов приложения по вычислительным узлам платформы (устройствам и ресурсам), это требует внесения изменений в модель (а, возможно, и ее существенной переработки), так как взаимодействие с ФБ, находящимися в другом ресурсе, производится с помощью специальных сервисных ФБ, т.е. модель зависит от структуры платформы.

### Выводы

IEC 61499 представляет собой развивающийся и перспективный стандарт описания вычислений в распределенных системах управления. В последнее время, с момента его принятия в 2005 г., к нему проявляется повышенный интерес: он активно обсуждается на конференциях, создан консорциум для его поддержки и развития, появляются различные инструменты автоматизированного проектирования в рамках стандарта. Основными достоинствами стандарта являются акцент на платформенную независимость (независимость от программно-аппаратной реализации) и лежащая в его основе распределенность вычислительного процесса, отличающая его от IEC 61131.

Тем не менее, несмотря на свою перспективность, стандарт имеет ряд существенных недостатков, препятствующих использованию в полной мере указанных достоинств.

Модель вычислений [3], предполагаемая IEC 61499, допускает блокировку части сети ФБ или даже одного простого ФБ. В то же время анализ возможности такой блокировки для простых ФБ связан с проблемами верификации управляющего автомата, представляющими собой отдельное актуальное направление исследований.

Наличие трех различных типов ФБ с различной степенью детализации поведения, а также другие приведенные выше особенности модели вычислений делают практически невозможным системный анализ моделей, описанных в рамках стандарта.

Не решено множество вопросов координации распределенных вычислений. В частности, допускается такая тривиальная ситуация, как нестабильность данных на входах ФБ. Не накладывается никаких ограничений на коммуникационную подсистему. Это приводит к необходимости ручной реализации протоколов, обеспечивающих должный уровень ее надежности. В то же время надежность и предсказуемость коммуникационной подсистемы являются часто неотъемлемыми требованиями при проектировании распределенных систем управления [7], и средства для их обеспечения можно было бы включить в стандарт.

Не обсуждаются требования реального времени и не предлагается никаких средств для их анализа в процессе проектирования. Модель вычислений IEC 61499 фактически делает их анализ невозможным, так как для простых ФБ нет ограничений на число выполняемых алгоритмов (пусть даже с известными WCET) и генерируемых событий и данных. Соответственно, в общем случае невозможно предсказать время его реакции, интенсивность генерации данных и другие временные характеристики. Анализ временных характеристик составных ФБ также невозможен, так как в них могут присутствовать сервисные ФБ, для которых вообще не существует каких-либо ограничений.

Отсутствие средств учета требований реального времени в стандарте описания систем управления, которые в основном являются системами реального времени, значительно снижает ценность этого стандарта.

Наличие указанных выше и многих других недостатков заставляет членов консорциума O<sup>3</sup>NEIDA вести исследования по его дополнительной конкретизации и разработке «профилей совместимости» (compliance profile), в которых решаются многие из присутствующих в IEC 61499 вопросов. В то же время предлагаемые решения [2, 5, 6], в

основном, касаются вопросов реализации проекта на вычислительной платформе, тогда как многие концептуальные вопросы модели вычислений, лежащей в основе стандарта, остаются до сих пор открытыми. Актуально решение этих вопросов – в первую очередь, вопросов учета требований реального времени и анализа временных характеристик сетей ФБ.

С этой точки зрения подход, предложенный в [4] для анализа схожей модели вычислений, выглядит перспективным. Подход основан на формальном описании вычислительного процесса, при котором его временные характеристики учитываются с самого начала (лежат в основе понятий события, сигнала и ФБ), на использовании для этих целей соответствующего математического аппарата и на унификации понятий «простой (атомарный) ФБ» и «составной ФБ». Это позволяет рассматривать иерархические модели на различных уровнях абстракции и заменять отдельные сети «простыми» ФБ, не прибегая к введению «вспомогательных» компонентов. Также это позволяет вычислять временные характеристики отдельных ФБ и всей системы, такие как время реакции и задержка управления, анализировать такие системные свойства, как детерминизм и живучесть. Разработка соответствующего «профиля совместимости» IEC 61499 с учетом такого подхода может решить указанные проблемы стандарта.

### Литература

1. International Electrotechnical Commission. IEC 61499. Function Blocks for Industrial Process Measurement and Control Systems. Part 1 – Architecture. – Geneva, 2005
2. Dubinin V., Vyatkin V. Towards a Formal Semantic Model of IEC 61499 Function Blocks // IEEE International Conference on Industrial Informatics, 16–18 August, 2006. – P. 6–11.
3. Лукичев А.Н. Денотативно-объектная модель вычислений для встроенных систем: дис. ... канд. техн. наук. – СПб: СПбГУ ИТМО, 2008 [Электронный ресурс]. – Режим доступа: [http://fppo.ifmo.ru/file/dis/a\\_68.pdf](http://fppo.ifmo.ru/file/dis/a_68.pdf), своб.
4. Лукичев А.Н. Вычисление временных характеристик объектно-событийных моделей встроенных систем // Научно-технический вестник СПбГУ ИТМО. – 2008. – Вып. 46. – С. 62–68.
5. Schwab C., Tangermann M., Lueder A., Kalogeras A., Ferrarini L. Mapping of IEC 61499 Function Blocks to Automation Protocols within the TORERO Approach // Proceedings of the IEEE International Conference on Industrial Informatics. – 2004. – P. 149–154.
6. Ferrarini L., Veber C. Implementation Approaches for the Execution Model of IEC 61499 Applications // Proceedings of the IEEE International Conference on Industrial Informatics. – 2004. – P. 612–617.
7. Постников Н.П. Динамические инструментальные компоненты // Научно-технический вестник СПбГИТМО(ТУ). Информационные, вычислительные и управляющие системы. – 2002. – № 6. – С. 83–88.

*Лукичев Александр Николаевич* – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кандидат технических наук, ассистент, [sash1@mail333.com](mailto:sash1@mail333.com)