

## Эволюция графовых моделей в АСУТП от функциональных блочных диаграмм к модульным системам обработки данных на основе направленных ациклических графов

С.Л. Власов

*Российский государственный университет имени А. Н. Косыгина (Технологии.  
Дизайн. Искусство)*

**Аннотация:** В статье рассматривается развитие графовых методов описания процессов обработки технологических данных в АСУТП и обоснована целесообразность перехода от традиционной парадигмы функциональных блочных диаграмм к архитектурам, основанным на направленных ациклических графах (англ. Directed Acyclic Graph — DAG). Показано, что использование DAG-модели обеспечивает расширенные возможности формализации, динамической конфигурации и распределённого исполнения вычислительных процессов, что позволяет устранять ограничения, присущие классическим средствам программирования программируемых логических контроллеров. Анализируется применимость модульных DAG-ориентированных систем в промышленной среде на примере архитектуры Tessera-DFE, реализующей загрузку графа, подключение модулей, событийное выполнение и независимые контексты обработки. Отмечено, что подобные решения способствуют повышению надёжности, масштабируемости и адаптивности программного обеспечения АСУТП в условиях роста объёмов данных, усложнения логики обработки и необходимости интеграции с внешними сервисами.

**Ключевые слова:** АСУТП, технологические данные, направленный ациклический граф, функциональные блочные диаграммы, ИЕС 61131-3, модульная архитектура, динамическая конфигурация, распределённое исполнение, плагинные модули, обработка событий, масштабируемость, надёжность, Tessera-DFE.

### Введение

Современные АСУТП характеризуются постоянным ростом объёмов данных, усложнением логики обработки сигналов и расширением требований к гибкости программных решений. Традиционные подходы, основанные на жёстко фиксированных последовательностях обработки и статично сконфигурированных алгоритмах, постепенно утрачивают способность эффективно поддерживать динамически изменяющиеся условия эксплуатации, рост числа источников данных и потребность в масштабируемой маршрутизации потоков информации [1]. В этих условиях возрастает значимость архитектур, обеспечивающих модульность, динамичность и формализованное представление логики обработки.

Исторически в промышленной автоматике широко применяется графовая парадигма описания логики в виде взаимосвязанных функциональных блоков (англ. Function Block Diagram — FBD) [2] согласно стандарту IEC 61131-3 (стандарт IEC 61131-3:2013. Программируемые контроллеры. Часть 3: Языки программирования. — Женева: Международная электротехническая комиссия, 2013.). Такая модель позволяет представлять алгоритм управления как ориентированный граф вычислений, в котором вершины соответствуют операциям преобразования данных, а ребра — передаче сигналов между ними. Несмотря на широкое распространение, например для разработки программного обеспечения ПЛК (программируемый логический контроллер — ПЛК), FBD ограничен статической природой конфигурации, фиксированным набором блоков и отсутствием механизмов динамического изменения структуры обработки. Эти особенности затрудняют адаптацию FBD-контуров к современным задачам высокоуровневой аналитики, интеграции с внешними сервисами, а также выполнению сложных многошаговых преобразований данных [3].

Переход к моделированию процессов обработки технологической информации на основе направленных ациклических графов (англ. Directed Acyclic Graph — DAG) представляет собой естественное развитие концепции функциональных блоков, сохраняя её наглядность, но существенно расширяя возможности по организации вычислений. DAG-модели позволяют формализовать произвольные цепочки преобразований, включая фильтрацию, агрегацию, обогащение, условную маршрутизацию и асинхронные вычисления. В отличие от FBD, ориентированные графы обработки могут строиться динамически, обновляться без остановки системы, а также использовать подключаемые модули, реализующие специализированные алгоритмы непредусмотренные базовой конфигурацией [4].

---

## Применение модульных DAG-систем в контуре АСУТП

Применение модульных DAG-ориентированных систем в контуре обработки технологических данных АСУТП приобретает актуальность в связи с возрастанием требований к интеграции с промышленными протоколами, системами мониторинга, аналитическими модулями и внешними вычислительными сервисами, что формирует потребность в универсальных программных платформах, сочетающих формализованное графовое описание процессов обработки с возможностью расширения функциональности за счёт подключаемых модулей. В качестве практического примера такой архитектуры рассматривается система Tessera-DFE [5], обеспечивающая исполнение DAG-структур, загрузку плагинов, динамическую конфигурацию последовательностей обработки и распределённое управление вычислительными узлами. Внедрение модульных DAG-моделей создаёт основу для разработки адаптивных, масштабируемых и формально определённых систем, способных изменять структуру обработки в соответствии с изменяющимися требованиями технологического процесса; в данной работе анализируется эволюционный переход от классической парадигмы функциональных блоков к универсальным DAG-ориентированным архитектурам и оценивается применимость подобного подхода на примере Tessera-DFE с акцентом на вопросы модульности, динамического формирования графа и интеграции с промышленными источниками данных [5].

Дополнительно следует отметить, что использование DAG-ориентированных моделей в АСУТП позволяет формализовать не только линейные цепочки обработки сигналов, но и сложные ветвящиеся сценарии с условиями, параллельным выполнением и агрегацией данных, что особенно важно при построении многоуровневых контуров мониторинга и диагностики. Модульный характер таких систем упрощает сопровождение и

развитие программного комплекса за счёт изоляции логики отдельных этапов обработки, повторного использования компонентов и снижения связности между ними, а также повышает устойчивость к изменениям состава оборудования и протоколов обмена. В перспективе данный подход создаёт технологическую основу для внедрения интеллектуальных методов анализа, включая потоковую аналитику и элементы машинного обучения, без необходимости переработки базовой архитектуры системы, что делает DAG-ориентированные платформы стратегически целесообразным решением для долгосрочного развития современных АСУТП.

### **Сопоставление функциональных диаграмм FBD с направленными ациклическими графами**

Графовые модели являются одним из базовых средств описания логики обработки технологических данных в АСУТП, поскольку обеспечивают формализованное представление взаимосвязей между операциями преобразования сигналов. Наиболее распространённые нотации стандарта ИЕС 61131-3, включая FBD, LD (англ. Ladder Diagram — LD; рус. релейно-контактные схемы / язык релейной (лестничной) логики) и SFC (англ. Sequential function chart — SFC; рус. последовательная функциональная диаграмма) реализуют ориентированные графы, в которых вершины соответствуют функциональным блокам, а ребра определяют направление передачи данных. Несмотря на широкое применение, данные модели характеризуются ограниченным набором доступных операций, статической структурой, отсутствием механизмов динамической реконфигурации и жёсткой привязкой исполнения к конкретному контроллеру. Эти особенности снижают их применимость в условиях возрастающих требований к адаптивности, масштабируемости и интеграции с внешними сервисами обработки данных [6].

Сопоставление функциональных диаграмм FBD с направленными ациклическими графами DAG демонстрирует, что FBD представляет собой частный случай более общей графовой формализации. В отличие от FBD, DAG-модели допускают произвольное определение типов узлов, обеспечивают корректность вычислений за счёт отсутствия циклов и допускают выполнение топологической сортировки [7]. Кроме того, DAG-подход поддерживает динамическое изменение структуры графа, подключение новых вычислительных модулей, перераспределение нагрузки и выполнение в распределённой вычислительной среде. Указанные свойства делают DAG-ориентированные модели перспективным направлением развития графовых средств описания технологической обработки данных в современных АСУТП.

В широкой практике обработки больших данных распределённые модели вычислений интегрируют графовые представления вычислений с автоматическим планированием и управлением исполнением. Ярким примером является модель MapReduce, предложенная J. Dean и S. Ghemawat [8], в которой обработка больших наборов данных выражается через две функции — Map (рус. Карта) и Reduce (рус. Уменьшать) — и автоматически распараллеливается и распределяется по кластерам машин, скрывая детали разделения данных, планирования задач и устойчивого выполнения в условиях отказов. Такая модель обработки, хотя и неявно представлена в виде DAG, использует этапы обработки, которые логически соответствуют узлам и зависимостям вычислительного графа, и показывает, как абстракции графовой обработки могут применяться на практике в масштабируемых системах.

### **Расширение выразительных возможностей модели.**

Архитектура системы Tessera-DFE представляет собой модульный программный комплекс, реализующий исполнение DAG-графов обработки

технологических данных и включающий подсистемы загрузки конфигурации, планирования вычислений и управления изолированными модулями. Граф обработки формируется на основе внешнего описания и загружается в исполнительное ядро, осуществляющее топологическую сортировку и распределение узлов по доступным планировщикам, поддерживающим различные модели активации, включая периодическое, событийное и немедленное выполнение. Каждая вычислительная операция реализуется в виде подключаемого модуля, исполняемого в собственном контексте, что обеспечивает изоляцию, возможность обновления без остановки системы и независимую обработку ошибок [7]. Поддержка событийной модели позволяет системе реагировать на изменения состояния источников данных и внешних сервисов, а модульная структура обеспечивает адаптацию графа под различные технологические процессы, включая фильтрацию, агрегирование, преобразование и маршрутизацию информации в производственных контурах.

Эволюционное сопоставление традиционных функциональных диаграмм FBD и DAG-ориентированных архитектур демонстрирует, что переход к DAG является логическим продолжением развития графовых методов описания вычислительных процессов в АСУТП. Сохранению подлежит базовая идея представления алгоритма в виде графа взаимосвязанных операций, обеспечивающего наглядность и формализованность обработки данных. Однако фундаментальные изменения связаны с расширением выразительных возможностей модели, включающим динамическое построение и изменение структуры графа, отсутствие ограничений на типы вычислительных узлов, поддержку распределённого исполнения и интеграцию с внешними источниками и потребителями данных. DAG-подход устраняет ограничения FBD, связанные со статичностью, фиксированным набором блоков и привязкой к одному

---

исполнительному устройству, обеспечивая тем самым основу для построения современных масштабируемых и адаптивных систем обработки технологической информации.

Развитие DAG-ориентированных архитектур в контексте АСУТП также согласуется с общими принципами построения надёжного и отказоустойчивого программного обеспечения, сформулированными в работах по инженерии промышленных систем. В частности, отмечается, что повышение надёжности достигается за счёт модульной структуры, изоляции компонентов, минимизации взаимных зависимостей и применения формальных моделей исполнения, что снижает вероятность распространения ошибок и обеспечивает контролируемое управление состоянием системы [9]. Эти положения коррелируют с концепцией DAG, где выполнение узлов допускает локальную обработку ошибок, независимо управляемые контексты и детерминированность порядка вычислений. Исследования в области распределённых графовых систем обработки данных также подтверждают, что применение DAG позволяет обеспечивать как структурированность вычислений, так и устойчивость к частичным отказам за счёт строгой формализации зависимостей и возможности переназначения вычислений в распределённой среде. Таким образом, использование DAG-модели в архитектуре Tessera-DFE соответствует современным требованиям к надёжности, масштабируемости и адаптивности программных решений для промышленных систем управления.

### **Выводы**

Представленный анализ показывает, что направленные ациклические графы являются современным обобщением традиционной парадигмы функциональных блоков, обеспечивая более широкие возможности формализации, динамического изменения и распределённого исполнения процессов обработки технологических данных. Ключевыми

---



характеристиками таких систем выступают модульность, масштабируемость и способность интегрироваться с разнообразными источниками и потребителями информации, что особенно актуально в условиях необходимости обработки данных в реальном времени и выявления отклонений в технологических процессах, отмеченной в современных исследованиях по промышленной автоматизации [10]. Архитектура Tesseract-DFE демонстрирует практическую реализуемость данных принципов, объединяя гибкую модель представления графа, механизм подключения модулей и поддержку различных сценариев выполнения, что подтверждает применимость DAG-подхода для построения современных адаптивных программных средств АСУТП.

Таблица № 1

Разница между FBD и ПО реализующее обработку DAG-модели

Характеристика	FBD / ПЛК-граф	DAG-движок (e.g. Tesseract-DFE)
Статичность	полностью статичен	может меняться динамически
Масштаб	граф в пределах ПЛК	распределённая система
Модули	фиксированные блоки	плагиновые модули, hot-swap
Ресурсы	ПЛК-лимит	кластер, контейнеры
Логика	простая	сложные цепочки обработки
Формат	IEC 61131-3	произвольный JSON/YAML/DAG-DSL
Выполнение	scan-cycle	cron, event, reactive, batch

### Литература

1. Rieger C., Ray I., Zhu Q., Haney M.A. (eds.) Industrial Control Systems Security and Resiliency: Practice and Theory. — Cham: Springer, 2019. — 276 p.
2. Петров И. В. Программируемые контроллеры. Стандартные языки и приёмы прикладного проектирования / Под ред. проф. В. П. Дьяконова. — М.: СОЛОН-Пресс, 2004. — 256 с. — ISBN 5-98003-079-4.
3. Nicolás M., Capel M., Adell C., García M. Evaluation of a UML-Based Versus an IEC 61131-3-Based Software Engineering Approach for Teaching PLC



---

Programming // IEEE Transactions on Education, 2013, Vol. 56, No. 3. P. 329–335. DOI: 10.1109/TE.2012.2226035.

4. Власов С.Л. Модульный программный подход к построению систем обработки технологических данных в АСУТП на основе направленных ациклических графов // Инженерный вестник Дона, 2025, №8. URL: ivdon.ru/ru/magazine/archive/n8y2025/10293.

5. Официальная документация проекта Tessera Data Flow Engine. — 2025. URL: [github.com/byzatic/Tessera-DFE/blob/main/README.md](https://github.com/byzatic/Tessera-DFE/blob/main/README.md). — Дата обращения: 22.11.2025.

6. Bang-Jensen J., Gutin G. Digraphs: Theory, Algorithms and Applications. 2nd ed. — Berlin: Springer, 2009. — 754 p.

7. Мякишев Д. В. Разработка программного обеспечения АСУ ТП на основе объектно-ориентированного подхода. — М.: Инфра-Инженерия, 2019. — 128 с. — ISBN 978-5-9729-0305-4.

8. Dean J., Ghemawat S. 6th Symposium on Operating Systems Design and Implementation (MapReduce: Simplified Data Processing on Large Clusters. (OSDI '04) December 6–8, 2004 San Francisco, CA, USA. URL: [static.googleusercontent.com/media/research.google.com/ru//archive/mapreduce-osdi04.pdf](https://static.googleusercontent.com/media/research.google.com/ru//archive/mapreduce-osdi04.pdf)

9. Мякишев Д. В. Принципы и методы создания надёжного программного обеспечения АСУТП. — М.: Инфра-Инженерия, 2017. — 114 с. — ISBN 978-5-9729-0179-1.

10. Фрасын П.Г. Анализ данных в реальном времени и обнаружение аномалий для промышленной автоматизации // Сборник научных статей III Международной студенческой междисциплинарной научно-практической конференции «Студенческая научно-исследовательская лаборатория: современное состояние и перспективы». — Краснодар: Академия маркетинга и социально-информационных технологий - ИМСИТ, 2024. — С. 907-911.

---

## References

1. Rieger C., Ray I., Zhu Q., Haney M.A. (eds.) Industrial Control Systems Security and Resiliency: Practice and Theory. Cham: Springer, 2019. 276 p.
2. Petrov I. V. Programmiruemy'e kontrollery'. Standartny'e yazy'ki i priyomy' prikladnogo proektirovaniya. Pod red. prof. V. P. D'yakonova. [Programmable Controllers. Standard Languages and Applied Design Techniques]. M.: SOLON-Press, 2004. 256 p. ISBN 5-98003-079-4.
3. Nicolás M., Capel M., Adell C., García M. IEEE Transactions on Education, 2013, Vol. 56, No. 3. pp. 329–335. DOI: 10.1109/TE.2012.2226035.
4. Vlasov S.L. Inzhenernyj vestnik Dona, 2025, №8. URL: [ivdon.ru/ru/magazine/archive/n8y2025/10293](http://ivdon.ru/ru/magazine/archive/n8y2025/10293).
5. Oficial'naya dokumentaciya proekta Tessera Data Flow Engine. [Official documentation of the Tessera Data Flow Engine project.] 2025. URL: [github.com/byzatic/Tessera-DFE/blob/main/README.md](https://github.com/byzatic/Tessera-DFE/blob/main/README.md). (accessed: 22.11.2025)
6. Bang-Jensen J., Gutin G. Digraphs: Theory, Algorithms and Applications. 2nd ed. Berlin: Springer, 2009. 754 p.
7. Myakishev D.V. Razrabotka programmnoho obespecheniya ASU TP na osnove ob''ektno-orientirovannogo podkhoda [Development of Software for APCS Based on Object-Oriented Approach]. M.: Infra-Inzheneriya, 2019. 128 p. ISBN 978-5-9729-0305-4.
8. Dean J., Ghemawat S. 6th Symposium on Operating Systems Design and Implementation 2004. OSDI '04. San Francisco, CA, USA. URL: [static.googleusercontent.com/media/research.google.com/ru//archive/mapreduce-osdi04.pdf](http://static.googleusercontent.com/media/research.google.com/ru//archive/mapreduce-osdi04.pdf)
9. Myakishev D.V. Printsipy i metody sozdaniya nadezhnogo programmnoho obespecheniya ASUTP [Principles and Methods of Creating Reliable Software for APCS]. M.: Infra-Inzheneriya, 2017. 114 p. ISBN 978-5-9729-0179-1.
10. Frasy'n P.G. Sbornik nauchny'x statej III Mezhdunarodnoj studencheskoj mezhdisciplinarnoj nauchno-prakticheskoy konferencii



«Studencheskaya nauchno-issledovatel'skaya laboratoriya: sovremennoe sostoyanie i perspektivy». Krasnodar: Akademiya marketinga i social'no-informacionny'x tehnologij. IMSIT, 2024. Pp. 907-911.

**Дата поступления: 2.12.2025**

**Дата публикации: 7.02.2026**