Модульный программный подход к построению систем обработки технологических данных в АСУТП на основе направленных ациклических графов

С.Л. Власов

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

Аннотация: В статье рассматривается метод построения программной системы для обработки технологических данных в составе автоматизированных систем управления технологическими процессами (АСУТП). Основу подхода составляет формализованное представление структуры обработки данных в виде направленного ациклического графа, где каждая вершина соответствует этапу преобразования информации, реализуемому с использованием подключаемых программных модулей. Предложена архитектура, обеспечивающая модульность, масштабируемость и повторное использование компонентов при организации обработки и маршрутизации данных. Выполнен анализ применимости данного подхода для построения гибких и адаптивных программных решений, ориентированных на эксплуатацию в условиях изменяющихся требований к логике обработки технологических сигналов.
Ключевые слова: АСУТП, обработка технологических данных, направленный ациклический граф, модульная архитектура, программные модули, отказоустойчивость, масштабируемость, динамическая конфигурация, интеграция в инфраструктуру, формализация обработки.

Введение

Современные автоматизированные системы управления технологическими процессами (АСУТП) характеризуются высоким уровнем сложности, обусловленным необходимостью обработки значительных объёмов разнородных технологических данных в реальном времени. При этом возрастают требования к гибкости программной архитектуры, адаптивности логики обработки сигналов, а также возможности оперативной модернизации без существенного вмешательства в функционирующие системы. В условиях динамически изменяющихся производственных задач и широкого разнообразия оборудования становится актуальной задача

разработки программных решений, обеспечивающих структурную модульность, масштабируемость и устойчивость к изменениям требований.

На практике в АСУТП по-прежнему нередко применяются монолитные или жёстко фиксированные программные схемы, плохо поддающиеся модернизации и масштабированию. Это затрудняет адаптацию систем к новым условиям эксплуатации, влечёт за собой необходимость значительных трудозатрат на внесение изменений, а также создаёт риски нарушения целостности системы при её доработке. В этой связи представляется целесообразным применение формализованных моделей, обеспечивающих строгое описание логики обработки данных и допускающих динамическое расширение функциональности.

В настоящей работе предлагается модульный программный подход к построению систем обработки технологических данных на основе направленного ациклического графа [1]. В рамках данной модели обработка информации представляется в виде графовой структуры, вершины которой реализуют последовательные этапы преобразования данных, а рёбра определяют маршруты передачи информации между модулями. Каждый этап обработки выполняется в рамках независимого программного компонента, подгружаемого динамически, что обеспечивает высокую степень изолированности и повторного использования элементов системы.

Целью настоящей работы является разработка архитектурного решения и формализованной модели обработки данных, ориентированной на применение в составе АСУТП, обеспечивающей гибкость логики исполнения, надёжность функционирования и возможность интеграции с существующими системами. В ходе исследования выполнен анализ применимости предложенного подхода, разработана прототипная реализация и проведены экспериментальные исследования, подтверждающие корректность и эффективность предложенной модели.

Анализ существующих подходов к обработке технологических данных

Современные автоматизированные системы управления технологическими процессами (АСУТП) характеризуются высокой степенью распределенности, интеграции разнородных источников информации и необходимости обработки больших объемов технологических данных в реальном времени. Обработка таких данных включает в себя процессы сбора, фильтрации, нормализации, агрегации, хранения и последующего анализа с целью поддержки принятия решений и автоматического управления [1].

Наиболее широко применяются архитектуры на основе OPC (OLE for Process Control) [2], обеспечивающие стандартизированный обмен данными между источниками и потребителями в АСУТП. В частности, спецификации OPC UA активно используются как в промышленных, так и в киберфизических системах, предоставляя средства абстрагирования от конкретного оборудования и поддерживая иерархическое представление данных. При этом необходимо отметить, что несмотря на высокую степень зрелости решений, реализация OPC-интерфейсов зачастую сопряжена с высокой стоимостью и необходимостью использования проприетарных решений.

Параллельно развиваются подходы, базирующиеся на использовании SCADA-систем с развитым программным обеспечением для обработки сигналов и событий. Эти решения, как правило, реализуют статические сценарии обработки, что ограничивает гибкость при изменении технологических требований. Также имеют место сложности с масштабируемостью и повторным использованием компонентов в условиях модификации технологического процесса. Подобные проблемы масштабируемости и адаптации архитектуры при модернизации также рассматриваются в работе Фрасына П. Г. “Математическая модель управления конфигурацией программных средств в автоматизированных

системах управления технологическими процессами” [3], где подчёркивается необходимость структурного разнесения логики обработки и ввода-вывода.

В научной литературе представлены различные модели описания процессов обработки данных в АСУТП. Одним из устойчивых направлений является использование графовых структур, в частности направленных ациклических графов, позволяющих формализовать последовательности обработки данных, отразить зависимости между шагами трансформации и обеспечить параллельное выполнение операций. При этом ноды графа могут соответствовать конкретным функциям обработки, а дуги описывать потоки данных.

В состав обработки данных в АСУТП входят не только функциональные преобразования, но и задачи согласования форматов, синхронизации времени и обеспечения устойчивости в условиях частичных отказов оборудования. ГОСТ 34.201–89 определяет принципы построения информационных технологий в автоматизированных системах, включая требования к архитектуре, взаимодействию подсистем и формализации обмена.

Также можно отметить растущую роль гибридных архитектур, сочетающих централизованное управление и локальную обработку данных, в том числе на базе промышленных контроллеров с поддержкой современных языков программирования, таких как Java и C++, что расширяет возможности по построению настраиваемых, модульных решений [5].

Это позволяет реализовать архитектуру, соответствующую требованиям базовой функциональности сбора и обработки технологических данных, но зачастую ограничены в части масштабируемости, повторного использования компонентов, адаптации к изменяющимся требованиям и возможности формального описания процессов обработки.

Формализация модели обработки данных на основе направленного ациклического графа

С целью повышения гибкости архитектуры программного обеспечения для автоматизированных систем управления технологическими процессами (АСУТП) в настоящей работе предлагается формализованная модель обработки данных, основанная на использовании направленного ациклического графа. Такая модель позволяет представить логическую структуру обработки технологических данных в виде взаимосвязанных узлов (вершин), выполняющих преобразование информации, и направленных связей (рёбер), определяющих последовательность передачи данных между ними.

Формально модель обработки данных можно представить как упорядоченную тройку [1]:

$$G = \langle V, E, P \rangle$$

где

- $V = \{v_1, v_2, \ldots, v_n\}$ — множество вершин графа, каждая из которых соответствует отдельному этапу обработки данных;
- $E \subseteq V \times V$ — множество направленных рёбер, отражающих отношения зависимости между этапами;
- $P: V \to \mathcal{F}$ — отображение, сопоставляющее каждой вершине функцию обработки из множества допустимых операций $\mathcal{F}$, реализуемых в виде программных модулей.

Ацикличность графа гарантирует отсутствие циклических зависимостей между этапами обработки, что обеспечивает определённость порядка выполнения вычислений и упрощает реализацию механизма

управления исполнением. Данные, поступающие на вход системы, передаются по рёбрам графа от одной вершины к другой, проходя через заранее заданную последовательность модулей. Вершины могут иметь несколько входящих и исходящих рёбер, что позволяет реализовать как линейные, так и разветвлённые схемы обработки.

Каждая вершина v_i может быть реализована в виде изолированного программного компонента, динамически загружаемого во время выполнения системы. Такой подход способствует модульности и расширяемости архитектуры, а также снижает зависимость между элементами системы. Разработанный программный механизм позволяет регистрировать, инициализировать и исполнять модули в соответствии с топологией заданного графа без необходимости ручной модификации основного кода приложения.

Дополнительное преимущество предлагаемой модели заключается в возможности статического анализа графа перед выполнением: проверка на ацикличность, выявление неиспользуемых или повторяющихся модулей, оценка критических путей исполнения, а также валидация корректности маршрутов передачи данных [4].

Это позволяет реализовать архитектуру, соответствующую требованиям, формализованного описания обработки технологических данных с использованием направленного ациклического графа, позволяет повысить надёжность, предсказуемость и адаптивность программных решений в составе АСУТП, обеспечивая при этом необходимую структурную гибкость для внедрения новых функциональных блоков без нарушения логики исполнения уже функционирующих подсистем.

Архитектурные принципы построения программной среды обработки данных в составе АСУ

В современных автоматизированных системах управления технологическими процессами наблюдается тенденция к усложнению логики обработки данных и необходимости адаптации к изменениям в конфигурации объектов управления. В этих условиях особую значимость приобретает структурная организация программных средств, обеспечивающая воспроизводимость, модульность и устойчивость обработки информации. Предлагаемый архитектурный подход основывается на формализованном представлении логики обработки в виде направленного ациклического графа, где вершины соответствуют обособленным операциям или функциональным модулям, а рёбра — однозначным маршрутам передачи данных.

Такое представление позволяет описывать последовательности преобразований без введения циклических зависимостей, обеспечивая тем самым однозначность вычислений и возможность проведения статического анализа структуры до начала исполнения. Каждая операция обработки данных интерпретируется как независимый программный компонент, изолированный от остальных элементов системы. Это снижает взаимозависимость модулей, упрощает обновление отдельных компонентов, а также способствует повышению надёжности за счёт локализации возможных сбоев. Используемый подход основан на принципах объектно-ориентированного проектирования программного обеспечения АСУТП [6].

Управляющая логика системы может быть сформулирована как интерпретатор топологии графа, определяющий порядок и условия активации отдельных операций на основе их входных связей. При этом конфигурация логики обработки представляется в виде внешнего описания, не зависящего от кода отдельных компонентов, что создаёт возможности для

гибкого переопределения поведения системы без необходимости её перекомпиляции или реконфигурации вручную [7].

Особое внимание при разработке концепции архитектуры уделяется вопросам надёжности и отказоустойчивости. Поскольку автоматизированные системы функционируют в условиях ограниченной доступности ресурсов и потенциально неблагоприятных внешних воздействий, критически важно обеспечить устойчивость исполнения при сбоях отдельных модулей. В данной архитектуре предполагается наличие механизмов контроля корректности входных и выходных данных, валидации маршрутов прохождения информации, а также регламентов повторного исполнения операций при обнаружении отклонений [8]. Предусматривается регистрация ошибок и событий, что позволяет анализировать поведение системы в постфактум-режиме и повышать её предсказуемость на этапе эксплуатации. Концепция построения устойчивых вычислительных подсистем в логике предлагаемой архитектуры согласуется с рекомендациями по проектированию отказоустойчивых решений, приведёнными во втором томе справочника по АСУТП [9].

Архитектурные принципы, положенные в основу предлагаемого подхода, соответствуют требованиям к надёжным промышленным решениям. Они обеспечивают возможность расширения системы без нарушения целостности обработки, допускают масштабирование конфигурации путём добавления новых модулей и обеспечивают прозрачность логики функционирования как на этапе проектирования, так и в процессе технического сопровождения. Подобная архитектурная модель может быть применима при построении как централизованных, так и распределённых систем управления, при этом она сохраняет независимость от конкретной предметной области или типа технологических процессов, в которых применяется.

Интеграция в программную инфраструктуру АСУТП

Предложенный программный подход к обработке технологических данных может быть интегрирован в существующую программную инфраструктуру автоматизированных систем управления технологическими процессами без вмешательства в замкнутые контуры управления. Это достигается за счёт логической изоляции подсистем обработки, реализуемых в виде вычислительных модулей, не затрагивающих функционирование основной управляющей логики.

Вычислительные модули, представляющие собой структурно и функционально автономные компоненты, осуществляют обработку, агрегацию и маршрутизацию данных, поступающих из различных источников в пределах АСУТП. Эти модули могут быть адаптированы для взаимодействия с внешними и внутренними подсистемами посредством реализации широкого спектра стандартных промышленных протоколов, включая OPC UA, Modbus, MQTT, а также специализированных протоколов, применяемых на объектах конкретных отраслей. Благодаря использованию единого механизма абстрагирования ввода-вывода, модули сохраняют независимость от конкретных реализаций протокольного стека, что упрощает адаптацию решения к различным эксплуатационным условиям.

Практические аспекты реализации клиента промышленного протокола Modbus TCP с применением open-source инструментов, ориентированных на обработку данных и взаимодействие с SCADA- и диагностическими системами, подробно рассмотрены в работе Фрасына П. Г. “Методологические основы работы с протоколом Modbus TCP с примером на высокоуровневом языке программирования Python” [10].

Архитектурное разделение вычислительных процессов позволяет использовать предложенное решение как надсистемный компонент, осуществляющий поддержку анализа, диагностики и интеллектуальной

маршрутизации данных в режиме, не нарушающем требований по детерминированности и устойчивости, предъявляемых к основным исполнительным подсистемам управления. Такая организация облегчает включение новой функциональности в действующие схемы АСУТП и допускает последовательную модернизацию без остановки технологического процесса [11].

Выводы

Рассмотрен модульный программный подход к построению систем обработки технологических данных в составе автоматизированных систем управления технологическими процессами. В качестве базовой модели выбрано представление логики обработки в виде направленного ациклического графа, обеспечивающего строгое формализованное описание последовательности операций и допускающего аналитическое исследование свойств структуры до начала исполнения.

Показано, что архитектурные принципы, заложенные в основу предлагаемого решения, соответствуют требованиям промышленной эксплуатации в условиях изменяющихся производственных задач. Разделение вычислительных процессов на изолированные модули, реализующие обособленные этапы обработки, обеспечивает адаптивность системы, её масштабируемость и структурную устойчивость. Особое внимание уделено вопросам надёжности, в том числе за счёт встроенных механизмов контроля целостности данных, повторного исполнения и регистрации событий.

Обоснована возможность интеграции разработанного решения в существующую программную инфраструктуру АСУТП. Предусмотрена реализация поддержки стандартных промышленных протоколов на уровне вычислительных модулей, что обеспечивает совместимость с различными

источниками данных и исполнительными подсистемами без нарушения целостности замкнутых контуров управления.

Предложенный подход может рассматриваться как основа для создания расширяемых программных систем, предназначенных для обработки, маршрутизации и интеллектуальной трансформации технологических данных в составе современных промышленных комплексов. Намеченные принципы организации обработки создают предпосылки для разработки универсальных решений, устойчивых к изменениям и пригодных для применения в широком диапазоне отраслей.

Литература

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

2. OPC Foundation. OPC 10000-1: UA Part 1: Overview and Concepts / OPC Foundation. — 2015. URL: reference.opcfoundation.org/Core/Part1/v105/docs/. — Дата обращения: 15.04.2025.

3. Фрасын П.Г. Математическая модель управления конфигурацией программных средств в автоматизированных системах управления технологическими процессами // Инженерный вестник Дона, 2025, №3. URL: ivdon.ru/magazine/archive/n3y2025/9924.

4. Блинов И. Н., Романчик В. С. Java. Промышленное программирование: практическое пособие. — Минск: УниверсалПресс, 2007. — 704 с. — ISBN 978-985-6699-63-7.

5. Кормен Т. Х., Лейзерсон Ч. И., Ривест Р. Л., Стайн К. Алгоритмы: построение и анализ. 3-е изд. — М.: Диалектика, 2020. — 1328 с. — ISBN 978-5-907114-11-1.

6. 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.

7. Ефимов С. В., Пушкарев М. И., Фадеев А. С. Программное обеспечение автоматизированных систем управления технологическими процессами: учебное пособие. — Томск: Томский политехнический университет, 2015. — URL: portal.tpu.ru/SHARED/e/EFIMOV/Academic/subjects/Tab/ПОАСУТП-пособие.pdf. — Дата обращения: 20.04.2025.

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

9. Фёдоров Ю. Н. Справочник инженера по АСУТП: проектирование и разработка. Т. 2. — М.: Инфра-Инженерия, 2016. — 484 с. — ISBN 978-5-9729-0123-4.

10. Фрасын П.Г., Никитин Н.В., Масанов Д.В., Рыжкова Е.А. Методологические основы работы с протоколом Modbus TCP с примером на высокоуровневом языке программирования Python // Инженерный вестник Дона, 2023, №11. URL: ivdon.ru/magazine/archive/n11y2023/8785.

11. Нестеров А. Л. Проектирование АСУТП: методическое пособие. Кн. 1. — М.: ДЕАН, 2006. — 552 с. — ISBN 5-93630-530-9.

References

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

2. OPC Foundation. OPC 10000-1: UA Part 1: Overview and Concepts. OPC Foundation, 2015. URL: reference.opcfoundation.org/Core/Part1/v105/docs/. (accessed: 15.04.2025)

3. Frasyn P.G. Inzhenernyj vestnik Dona, 2025, №3. URL: ivdon.ru/magazine/archive/n3y2025/9924.

4. Blinov I.N., Romanchik V.S. Java. Promyshlennoe programmirovanie: prakticheskoe posobie [Java: Industrial Programming. A Practical Guide]. Minsk: UniversalPress, 2007. 704 s. ISBN 978-985-6699-63-7.

5. Kormen T.Kh., Lejzerson Ch.I., Rivest R.L., Stajn K. Algoritmy: postroenie i analiz [Algorithms: Construction and Analysis]. 3-e izd. M.: Dialektika, 2020. 1328 p. ISBN 978-5-907114-11-1.

6. Myakishev D.V. Razrabotka programmnogo 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.

7. Efimov S.V., Pushkarev M.I., Fadeev A.S. Programmnoe obespechenie avtomatizirovannykh sistem upravleniya tekhnologicheskimi protsessami: uchebnoe posobie [Software for Automated Process Control Systems: Study Guide]. Tomsk: Tomskij politehnicheskij universitet, 2015. URL: portal.tpu.ru/SHARED/e/EFIMOV/Academic/subjects/Tab/ПОАСУТП-пособие.pdf (accessed: 20.04.2025)

8. Myakishev D.V. Printsipy i metody sozdaniya nadezhnogo programmnogo obespecheniya ASUTP [Principles and Methods of Creating Reliable Software for APCS]. M.: Infra-Inzheneriya, 2017. 114 p. ISBN 978-5-9729-0179-1.

9. Fedorov Yu.N. Spravochnik inzhenera po ASUTP: proektirovanie i razrabotka [Engineer's Handbook on APCS: Design and Development]. T. 2. M.: Infra-Inzheneriya, 2016. 484 s. ISBN 978-5-9729-0123-4.

10. Frasyn P.G., Nikitin N.V., Masanov D.V., Ryzhkova E.A. Inzhenernyj vestnik Dona, 2023, №11. URL: ivdon.ru/magazine/archive/n11y2023/8785.

11. Nesterov A.L. Proektirovanie ASUTP: metodicheskoe posobie [Design of APCS: Methodological Guide]. Kn. 1. M.: DEAN, 2006. 552 p. ISBN 5-93630-530-9.

Дата поступления: 24.04.2025
Дата публикации: 25.07.2025