

Методологические основания выбора средств автоматизации сопровождения программной среды верхнего уровня автоматизированных систем управления технологическими объектами

П.Г. Фрасын, Н.В. Никитин

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

Аннотация: Программная среда верхнего уровня автоматизированных систем управления технологическими процессами требует воспроизводимого и предсказуемого сопровождения. В статье сформулированы требования к средствам автоматизации, обеспечивающим согласованное состояние среды исполнения и независимость процедур от ее текущего состояния. Проведена оценка архитектурных свойств распространенных решений. Обосновано совместное применение Terraform и Ansible как основы формализованной модели сопровождения.

Ключевые слова: программная среда, автоматизированная система управления, сопровождение, воспроизводимость, конфигурация, автоматизация, жизненный цикл, вычислительная среда, управление изменениями, язык программирования.

Введение

Программные компоненты верхнего уровня автоматизированных систем управления технологическими процессами (АСУТП) выполняют задачи мониторинга параметров, диспетчеризации режимов и производственного планирования. Их корректное функционирование возможно только при наличии согласованной среды исполнения, включающей вычислительные узлы, параметры размещения, взаимосвязи между службами и конфигурационные зависимости.

Нарушение согласованности среды исполнения с заданным техническим состоянием приводит к искажению работы компонентов, снижает воспроизводимость восстановительных процедур и ограничивает управляемость программной среды. При отсутствии формализованных процедур сопровождения отклонения накапливаются, что затрудняет точное воспроизведение условий исполнения, снижает устойчивость системы и

усложняет поддержание ее работоспособности при переходных и аварийных режимах.

Поддержание согласованного состояния среды исполнения требует применения средств, обеспечивающих машиночитаемое описание ее структуры, автоматическое обнаружение отклонений от эталонной конфигурации и воспроизводимое выполнение корректирующих действий [1]. Автоматизация таких процессов необходима для исключения ручных вмешательств, снижения вероятности накопления ошибок и обеспечения предсказуемого поведения системы на протяжении всего жизненного цикла [2].

Целью настоящей работы является установление применимости существующих технических решений к задачам сопровождения программной среды исполнения в системах верхнего уровня автоматизации. Такая среда должна обеспечивать воспроизводимость, устойчивость и техническую определенность операций при длительной эксплуатации компонентов АСУТП.

Задача исследования заключается в определении условий, при которых средства автоматизации могут быть использованы для сопровождения среды исполнения, с учетом требований воспроизводимости, независимости от текущего состояния, определенности результата и допустимости применения в условиях ограниченного доступа к вычислительным ресурсам.

Требования к средствам сопровождения среды исполнения

Организация воспроизводимого и управляемого сопровождения программной среды верхнего уровня требует использования средств, обеспечивающих технически определенные и контролируемые операции. Такие средства должны:

- обеспечивать представление структуры среды исполнения в машиночитаемой форме;

- позволять проверку соответствия текущего состояния утвержденному эталону;
- поддерживать автоматизированное выполнение действий по приведению среды в согласованное состояние.

Средства, применяемые в системах верхнего уровня автоматизации, должны удовлетворять следующим условиям:

1. Отделенность от прикладной логики. Средства сопровождения не должны изменять структуру или поведение компонентов верхнего уровня и не вмешиваться в реализуемые ими функции. Их область действия ограничивается подготовкой и поддержанием среды, в которой размещаются прикладные компоненты.

2. Идемпотентность операций. Повторное применение процедур сопровождения не должно нарушать корректно функционирующее состояние среды исполнения.

3. Возможность работы при ограниченном доступе. Сопровождение должно осуществляться без постоянного присутствия на объекте, с минимальной необходимостью ручного взаимодействия с вычислительными узлами.

4. Безагентное выполнение. Безагентным считается выполнение операций сопровождения без установки дополнительных программных компонентов (агентов) на управляемых устройствах. Такое выполнение особенно важно в случаях, когда допустимо только одностороннее взаимодействие со стороны управляющей среды.

5. Машиночитаемая фиксация состояния. Все действия сопровождения должны сопровождаться формализованной регистрацией, пригодной для аудита и последующего восстановления истории изменений [3].

6. Охват всего жизненного цикла. Применяемые средства должны быть пригодны как для инициализации среды исполнения, так и для сопровождения и обновления прикладных компонентов, обеспечивая воспроизводимость операций на всех этапах.

Обзор существующих средств автоматизации сопровождения среды исполнения

В современных автоматизированных системах управления технологическими процессами отсутствуют специализированные средства, обеспечивающие воспроизводимое и контролируемое сопровождение среды исполнения компонентов верхнего уровня в соответствии с требованиями, обусловленными распределенной структурой и длительным жизненным циклом. В связи с этим возникает необходимость анализа решений, применяемых в практике сопровождения информационных систем, архитектурные особенности которых потенциально соответствуют условиям воспроизводимости, независимости от текущего состояния и технической определенности выполняемых операций.

Наибольшее распространение получили инструменты, реализующие централизованное управление параметрами программной среды и фиксацию ее целевого состояния [4]. В настоящем разделе рассматриваются решения, получившие широкое применение в промышленной и корпоративной эксплуатации: Puppet, Ansible, Chef, Fabric, Terraform и SaltStack.

Puppet реализует централизованную архитектуру, в которой управляющий узел передает инструкции управляемым устройствам посредством встроенных служебных модулей [5]. Для функционирования системы требуется предварительная установка компонентов Puppet на каждом узле. Такая модель обеспечивает контроль исполнения операций, но затрудняет применение в условиях ограниченного или периодического

доступа, где невозможна установка дополнительных элементов в среду исполнения.

Ansible использует модель без установки агентных компонентов на управляемых устройствах. Все действия выполняются с управляющего узла по заранее заданным сценариям, с использованием стандартных сетевых протоколов [6]. Оценка текущего состояния среды и выполнение корректирующих действий осуществляются через идиempotentные модули, что допускает многократное применение сценариев без нарушения корректного функционирования.

Chef по архитектуре сходен с Puppet, но использует специализированный язык описания состояния, не совместимый с другими системами [7]. Для работы требуется установка служебных компонентов на каждом управляемом устройстве и постоянное сетевое соединение с управляющим сервером, что ограничивает применимость в слабо связанных и изолированных вычислительных средах.

Fabric представляет собой инструмент для удаленного выполнения команд, не предусматривающий формализованного описания состояния среды исполнения [8]. Он не включает механизмов фиксации параметров и не поддерживает воспроизводимость действий при повторных запусках. Такие инструменты допустимы для автоматизации отдельных операций, но не соответствуют требованиям сопровождения на протяжении всего жизненного цикла программной среды.

Terraform предназначен для автоматизированного создания и воспроизведения структуры вычислительной среды на основе формализованного описания. Модель его работы ориентирована на достижение заданного состояния всей среды, включая размещение вычислительных узлов, сетевых компонентов и служебных ресурсов [9]. Инструмент обеспечивает высокую степень воспроизводимости при

инициализации среды, но ограничен в задачах конфигурации уже функционирующих программных компонентов.

SaltStack ориентирован на масштабируемое управление конфигурацией, поддерживает как централизованную, так и агентную архитектуру [10]. Даже в минимальной конфигурации предполагается установка управляющего компонента на каждую целевую систему, а также наличие постоянной или периодической связи между узлами.

Сравнительная оценка применимости

Для сопоставления пригодности существующих средств автоматизации к задачам сопровождения программной среды верхнего уровня выполнен анализ их архитектурных свойств в контексте сформулированных требований. В основу анализа легли следующие характеристики: возможность выполнения без установки агентного программного обеспечения, поддержка воспроизводимости состояния среды, направленность на инфраструктурный или прикладной уровень, а также пригодность для использования при ограниченном доступе к вычислительным узлам. Основные характеристики инструментов приведены в таблице 1, построенной на основе сопоставления их архитектурных свойств с выделенными требованиями.

Таблица 1. Характеристики инструментов автоматизации сопровождения программной среды исполнения

Инструмент	Безагентность	Управление конфигурацией	Управление инфраструктурой	Применим при ограниченном доступе
Puppet	Нет	Да	Нет	Нет
Ansible	Да	Да	Ограниченно	Да
Chef	Нет	Да	Нет	Нет
Fabric	Да	Нет	Нет	Частично

Terraform	Да	Ограниченно	Да	Да
SaltStack	Нет	Да	Частично	Нет

Анализ показывает, что ни одно из рассмотренных решений в отдельности не удовлетворяет совокупности требований, предъявляемых к средствам сопровождения программной среды исполнения в распределённых системах верхнего уровня автоматизации. Вместе с тем, их совместное применение, в частности, комбинация Terraform и Ansible позволяет охватить полный цикл задач: от воспроизводимого развёртывания вычислительной среды до настройки и сопровождения прикладных компонентов.

Инструмент Terraform предназначен для построения структуры среды исполнения, включая размещение вычислительных узлов, сетевых связей и служебных зависимостей, с обеспечением контролируемого воспроизведения состояния. Ansible, в свою очередь, применяется для конфигурирования компонентов в рамках уже развёрнутой среды, не требует установки вспомогательного программного обеспечения на управляемых устройствах и поддерживает идемпотентность операций.

Обобщённая схема взаимодействия инструментов представлена на рисунке 1.

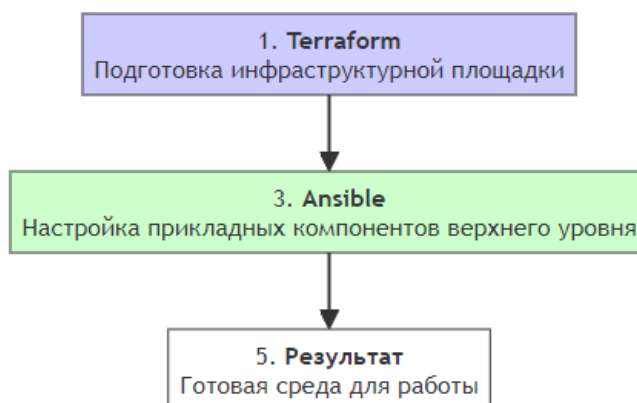


Рис. 1. – Взаимодействие средств автоматизации при сопровождении среды исполнения

Такое разграничение задач сопровождения между двумя средствами позволяет компенсировать их индивидуальные ограничения и применить каждое из них в наиболее подходящей зоне ответственности. В результате достигается воспроизводимость среды исполнения и управляемость её жизненного цикла в условиях ограниченного доступа и длительной эксплуатации [11].

Выводы

В статье рассмотрены принципы выбора средств автоматизации сопровождения программной среды исполнения в системах верхнего уровня автоматизации технологических процессов. Показано, что поддержание устойчивой и воспроизводимой работы программных компонентов требует согласованной среды исполнения, включающей параметры размещения, структуру вычислительных узлов и межсервисные связи.

Сформулированы требования к средствам автоматизации, обеспечивающим воспроизводимость операций, определённую результативность и независимость от текущего состояния среды. Выполнен обзор существующих инструментов сопровождения, проанализирована их пригодность для применения в распределённых программных системах.

Установлено, что комбинация средств Terraform и Ansible позволяет реализовать полную цепочку автоматизированного сопровождения — от развёртывания инфраструктуры до настройки прикладных компонентов — без вмешательства в их функциональную логику. Такое сочетание обеспечивает формализованный и управляемый жизненный цикл программной среды исполнения и может рассматриваться как практическая основа сопровождения в условиях ограниченного доступа и повышенной устойчивости.

Проведённый анализ может служить основой для разработки регламентированных методик сопровождения распределённых программных сред в автоматизированных системах управления.

Литература

1. Безпятый М.В. Автоматизация и оптимизация процессов разработки и развертывания в DevOps: применение современных методов и инструментов // Инновации и инвестиции. 2023. №7. С. 458-464. URL: cyberleninka.ru/article/n/avtomatizatsiya-i-optimizatsiya-protseessov-razrabotki-i-razvertyvaniya-v-devops-primenenie-sovremennyh-metodov-i-instrumentov (дата обращения: 28.04.2025)
2. Данилин А.О., Кол М.Д., Петрухнова Г.В. Мультипарадигмальность управления жизненным циклом дефектов программных и аппаратных систем // Вестник Воронежского государственного технического университета. 2015. URL: cyberleninka.ru/article/n/multiparadigmalnost-upravleniya-zhiznennym-tsiklom-defektov-programmnyh-i-apparatnyh-sistem (дата обращения: 28.04.2025)
3. Фрасын П.Г. Математическая модель управления конфигурацией программных средств в автоматизированных системах управления технологическими процессами // Инженерный вестник Дона. 2024. №4. URL: ivdon.ru/ru/magazine/archive/n3y2025/9924 (дата обращения: 04.05.2025).
4. Костромин Р.О. Сравнительный обзор средств управления конфигурациями ресурсов вычислительной среды функционирования цифровых двойников // Информационные и математические технологии в науке и управлении. 2021. №1(21). С. 132-145. URL: imt-journal.ru/archive/public/article?id=186 (дата обращения: 29.04.2025).
5. Воронцов Ю.А., Михайлова Е.К. Комбинированная архитектура системы конфигурационного управления вычислительной инфраструктурой // International Journal of Open Information Technologies. 2021. Т.9. №11. С. 71–

75. ISSN 2307-8162. URL: injoit.org/index.php/j1/article/view/1178 (дата обращения: 29.04.2025).

6. Фомин Д.С., Бальзамов А.В., Савкина А.В., Федосин С.А. Проблемы взаимодействия разработчиков с облачными инфраструктурами на базе Kubernetes в тестовых средах // Известия высших учебных заведений. Поволжский регион. Технические науки. 2023. №3. С. 31–40 URL: izvuz_tn_eng.pnzgu.ru/tn3323 (дата обращения: 29.04.2025).

7. Мещеряков С.В., Руденко А.О., Щемелинин Д.А. Международные конференции ASE по науке и компьютерной обработке больших данных // Научно-технические ведомости Санкт-Петербургского политехнического университета. Информатика. Телекоммуникации. Управление. 2015. № 1(212). С. 110–119. URL: mathnet.ru/php/archive.phtml?wshow=paper&jrnid=ntitu&paperid=98&option_lang=rus (дата обращения: 30.04.2025).

8. Androulaki E., Barger A., Bortnikov V., Cachin C., Christidis K., De Caro A., Enyeart D., Ferris C., Laventman G., Manevich Y., Muralidharan S., Murthy C., Nguyen B., Sethi M., Singh G., Smith K., Sorniotti A., Stathakopoulou C., Vukolić M., Cocco S. W., Yellick J. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains // arXiv preprint arXiv:1801.10228. 2018. URL: arxiv.org/abs/1801.10228 (дата обращения: 30.04.2025).

9. Howard M. Terraform – Automating Infrastructure as a Service // arXiv preprint arXiv:2205.10676. 2022. URL: arxiv.org/abs/2205.10676 (дата обращения: 30.04.2025)

10. Корягин С.А., Грицкевич Е.В. Разработка алгоритма организации инфраструктуры информационной безопасности с применением системы управления конфигурацией SaltStack при передаче данных по волоконно-оптической линии связи // Интерэкспо Гео-Сибирь 2019. Магистерская



научная сессия. Первые шаги в науке. Т.6. №2. 2019. С. 92-98. URL: geokniga.org/books/36687 (дата обращения: 30.04.2025)

11. Корнюшкин Д.А., Крылов А.А. Задачи совершенствования современных автоматизированных систем управления технологическими процессами // Инженерный вестник Дона. 2023. №3. URL: ivdon.ru/ru/magazine/archive/n3y2023/8284/ (дата обращения: 30.04.2025)

References

1. Bezpyatyi M.V. Innovatsii i investitsii. 2023. No. 7. pp. 458–464. URL: cyberleninka.ru/article/n/avtomatizatsiya-i-optimizatsiya-protseessov-razrabotki-i-razvertyvaniya-v-devops-primenenie-sovremennyh-metodov-i-instrumentov (accessed: 28.04.2025).

2. Danilin A.O., Kol M.D., Petrukhnova G.V. Vestnik Voronezhskogo gosudarstvennogo tekhnicheskogo universiteta. 2015. URL: cyberleninka.ru/article/n/multiparadigmalnost-upravleniya-zhiznennym-tsiklom-defektov-programmnyh-i-apparatnyh-sistem (accessed: 28.04.2025).

3. Frasyn P.G. Inzhenernyj vestnik Dona. 2024. No. 4. URL: ivdon.ru/ru/magazine/archive/n3y2025/9924 (accessed: 04.05.2025).

4. Kostromin R.O. Informatsionnye i matematicheskie tekhnologii v nauke i upravlenii. 2021. No. 1(21). pp. 132–145. URL: imt-journal.ru/archive/public/article?id=186 (accessed: 29.04.2025).

5. Vorontsov Yu.A., Mikhaylova E.K. International Journal of Open Information Technologies. 2021. Vol. 9. No. 11. pp. 71–75. ISSN 2307-8162. URL: injoit.org/index.php/j1/article/view/1178 (accessed: 29.04.2025).

6. Fomin D.S., Balzamorov A.V., Savkina A.V., Fedosin S.A. Izvestiya vysshikh uchebnykh zavedenij. Povolzhskij region. Tekhnicheskie nauki. 2023. No. 3. pp. 31–40. URL: izvuz_tn_eng.pnzgu.ru/tn3323 (accessed: 29.04.2025).

7. Meshcheryakov S.V., Rudenko A.O., Shchemelinin D.A. Nauchnotekhnicheskie vedomosti SPbPU. Informatika. Telekommunikatsii.



Upravlenie. 2015. No. 1(212). pp. 110–119. URL: mathnet.ru/php/archive.phtml?wshow=paper&jrnid=ntitu&paperid=98&option_lang=rus (accessed: 30.04.2025).

8. Androulaki E., Barger A., Bortnikov V., Cachin C., Christidis K., De Caro A., Enyeart D., Ferris C., Laventman G., Manevich Y., Muralidharan S., Murthy C., Nguyen B., Sethi M., Singh G., Smith K., Sorniotti A., Stathakopoulou C., Vukolić M., Cocco S.W., Yellick J. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. arXiv preprint arXiv:1801.10228. 2018. URL: arxiv.org/abs/1801.10228 (accessed: 30.04.2025).

9. Howard M. Terraform – Automating Infrastructure as a Service. arXiv preprint arXiv: 2205.10676. 2022. URL: arxiv.org/abs/2205.10676 (accessed: 30.04.2025).

10. Koryagin S.A., Gritskevich E.V. Interexpo Geo-Sibir'. Magistrskaya nauchnaya sessiya. Pervye shagi v nauke. 2019. Vol. 6. No. 2. pp. 92–98. URL: geokniga.org/books/36687 (accessed: 30.04.2025).

11. Korniyushkin D.A., Krylov A.A. Inzhenernyj vestnik Dona. 2023. No. 3. URL: ivdon.ru/ru/magazine/archive/n3y2023/8284/ (accessed: 30.04.2025).

Дата поступления: 5.05.2025

Дата публикации: 25.06.2025