ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ГОРНОМ ДЕЛЕ

УДК 004.05 DOI: 10.21440/0536-1028-2023-3-113-126

Разработка инфраструктуры для проведения юнит-тестирования программной реализации системы управления дегазационной вакуум-насосной станцией

Абдрахманов М. И.[1]*, Гардт Д. А.[1]
[1] Уральский государственный горный университет, г. Екатеринбург, Россия
e-mail: marat-ab@mail.ru

Реферат

Цель работы. *Разработка системы автоматизации тестирования программной реализации технологического алгоритма работы системы управления дегазационной вакуум-насосной станцией угольной шахты.*

Актуальность. *Вакуум-насосная станция является важнейшим компонентом дегазационной системы шахты, от корректности ее функционирования зависит работа системы дегазации. Ключевым компонентом системы автоматизации вакуум-насосной станции является программируемый логический контроллер с развернутым на нем соответствующим программным обеспечением. Очень важно, чтобы программная реализация технологического алгоритма работы системы была корректной. Для проверки соответствия реализации требованиям, как правило, применяется ручное тестирование по заранее составленному тест-плану, который формируется на основании требований к системе. Ручное тестирование имеет ряд недостатков: данный процесс имеет значительную длительность по времени; в ходе проведения тестирования возможны ошибки из-за человеческого фактора; такой тип тестирования возможен только при наличии тестового стенда и/или собранного образца системы управления. Требования к алгоритму могут быть изменены в процессе разработки программы, а также на этапах наладки и ввода в эксплуатацию, что значительно усложняет процесс тестирования.*

Результаты. *Для решения задачи автоматизации тестирования программной реализации технологического алгоритма необходимо изолировать технологический алгоритм управления от программных компонент, которые отвечают за работу с аппаратным обеспечением, и предоставить к нему доступ через стандартные интерфейсы (OPC UA). Дополнительное технологическое оборудование (задвижки и т. д.), используемое в системе автоматизации, следует реализовать в виде программных эмуляторов с возможностью задания их режимов работы. Создав дополнительно программный слой взаимодействия с технологическим алгоритмом и эмулятором на языке программирования с развитыми инструментами для тестирования (Python), можно реализовать сценарии тестирования из тест-плана на этом языке, что позволит выполнять тестирование не вручную, а в автоматическом режиме.*

Выводы. *Предлагаемый подход к автоматизации работы по тестированию программной реализации технологического алгоритма позволит значительно упростить процесс разработки программного обеспечения для такого вида систем автоматизации и управления.*

Ключевые слова: *дегазация; вакуум-насосная станция; автоматизация; тестирование; ручное тестирование; автоматизированное тестирование.*

Введение. На современных угольных шахтах одним из главных приоритетов является безопасность ведения горных работ. Значительная часть аварий, которые

происходят на угольных предприятиях, связана со взрывом метана. Метан – часть рудничного газа, который выделяется из угля и горных пород в горные выработки. Накапливаясь в закрытом помещении, смесь метана с воздухом может стать взрывоопасной [1, 2].

Одним из способов борьбы с метаном является дегазация угольных пластов [2, 3]. Дегазационная система шахты – это комплекс технических средств, предназначенный для изолированного от атмосферы шахты отвода метановоздушной смеси на поверхность. Важнейшим компонентом дегазационной системы является вакуум-насосная станция (ВНС), она создает вакуум, за счет которого смесь извлекается из угольного пласта или выработанного пространства и транспортируется на поверхность [4–7], (*Инструкция по дегазации угольных шахт (РД-15-09-2006). Сер. 05. Вып. 22. М.: Научно-технический центр по безопасности в промышленности, 2012. 250 с.*). От исправности функционирования ВНС зависит работоспособность всей системы дегазации.

Система автоматизации работы ВНС предполагает решение задач контроля, управления, регулирования и защиты в объеме, требуемом для конкретной установки [4].

Техническое задание на разработку такой системы включает требования, которые могут быть формализованы в виде набора алгоритмов. Полученные алгоритмы используются в дальнейшем для разработки программного обеспечения (ПО) для программируемого логического контроллера (ПЛК), являющегося центральным компонентом системы автоматизации.

Разработка ПО, как правило, ведется на этапе, когда еще нет готового аппаратного комплекса, включающего ПЛК, модули ввода/вывода и т. д. Это создает значительные трудности для тестирования корректности реализации алгоритма, которое в полном объеме может быть проведено только при наличии всех компонентов разрабатываемого комплекса.

В процессе разработки ПО часто приходится вносить изменения, которые могут затронуть уже созданные части программы. Это приводит к необходимости частого проведения полнофункционального тестирования, которое для такого типа решений выполняется преимущественно вручную, что приводит к значительным временным затратам.

Требования к алгоритму работы системы управления могут измениться в процессе разработки ПО, и даже во время наладки и ввода системы в эксплуатацию. Зачастую требуется доработка системы «по месту» из-за особенностей объекта автоматизации.

Поэтому нужен подход к тестированию ПО, который бы учитывал все эти особенности и позволял проводить тестирование реализации технологической части алгоритма системы управления в автоматическом режиме и без наличия аппаратного комплекса системы автоматизации.

В статье представлен предлагаемый нами подход к автоматизации тестирования реализации технологического алгоритма работы системы управления на примере дегазационной вакуум-насосной станции угольной шахты.

Алгоритм управления подземной дегазационной ВНС. В общем виде система автоматизации ВНС предполагает наличие следующих подсистем: контроль, управление, регулирование и защита. Системы автоматизации ВНС подземного типа (ПВНС), как правило, менее сложные по сравнению с ВНС наземного типа (*Инструкция по дегазации угольных шахт…*).

Среди управляющих воздействий, которые оказывает ПЛК через модули релейных выходов на элементы ПВНС, выделим следующие:

– разрешение на запуск вакуум-насоса 1 (реле замкнуто), остановка вакуум-насоса 1 (реле разомкнуто), реализуется через релейный выход 1 (DO1);

– разрешение на запуск вакуум-насоса 2 (реле замкнуто), остановка вакуум-насоса 2 (реле разомкнуто), реализуется через релейный выход 2 (DO2);

– сигнал «Авария», используется для сигнализации и/или передачи сторонним системам информации об аварийном состоянии, реализуется через релейный выход 3 (DO3).

Для управления подачей метановоздушной смеси в ПВНС используется поворотный затвор (далее ПЗ), который может управляться через интерфейс RS485 (протокол Modbus). Система дополнительно оборудована кнопочным постом для подачи дискретных сигналов в ПЛК для открытия/закрытия затвора.

В качестве ПЛК выберем вариант с возможностью программирования на языках из группы МЭК, например поддерживающий среду Codesys [8].

Выделим сокращенный набор требований к работе ПВНС:

– если концентрация метана в смеси на входе в ПВНС находится в диапазоне 3–25 %, то остановить насосы Н1 и Н2, активировать сигнал «Авария» и закрыть ПЗ;

– если концентрация метана в смеси на входе в ПВНС находится в диапазоне 0–3 % или больше 25 %, то подать разрешение на запуск насосов Н1 и Н2, деактивировать сигнал «Авария», если он был активен, и снять управление с ПЗ;

– если давление на выходе насоса Н1 выше верхней аварийной уставки, то активировать сигнал «Авария»;

– если давление на выходе насоса Н1 ниже верхней аварийной уставки, то деактивировать сигнал «Авария», если он был активен;

– если давление на выходе насоса Н2 выше верхней аварийной уставки, то активировать сигнал «Авария»;

– если давление на выходе насоса Н2 ниже верхней аварийной уставки, то деактивировать сигнал «Авария», если он был активен;

– если хотя бы один из насосов Н1 или Н2 был в работе, а в данный момент оба насоса остановлены, то закрыть ПЗ;

– если оба насоса были остановлены, и в данный момент насосы остановлены, то не оказывать воздействие на ПЗ;

– если ПЗ не открыт (угол открытия < 5 %), то остановить насосы Н1 и Н2 и заблокировать возможность их запуска, иначе подать разрешение на запуск насосов Н1 и Н2;

Формализуем эти требования в виде диаграммы видов деятельности UML [9, 10], которая представлена на рис. 1.

Особенности процесса разработки ПО для систем контроля и управления. Основные трудности, с которыми сталкивается разработчик ПО для систем контроля и управления, перечислены во введении. Также следует отметить, что в ряде технических средств, на базе которых разрабатываются системы автоматизации, нет инструментов для разработки тестов и запуска их в изоляции от аппаратного обеспечения. Как уже было отмечено, в этом случае для полноценного тестирования требуется собрать программно-аппаратный комплекс с соответствующим ПЛК и, при необходимости, стенд для тестирования. Тестирование в этом случае выполняется преимущественно вручную.

Тестирование реализации технологического алгоритма работы системы управления ПВНС. На этапе проектирования системы автоматизации, как правило, уже известно количество и тип входных и выходных сигналов, а также оборудование, управление которым будет производится. Исходя из этого выбирается тот или иной тип ПЛК и состав его модулей. При этом требования к алгоритму

управления могут быть не полными и в процессе работы над проектом могут измениться.

Это влияет на ПО для ПЛК такой системы. Условно в ПО можно выделить две части, одна из них изменяется редко, она связана с настройкой под конкретный состав модулей и набор сигналов. Вторая изменяется часто, в рамках нее реализуется технологический алгоритм, решающий задачи контроля, управления,

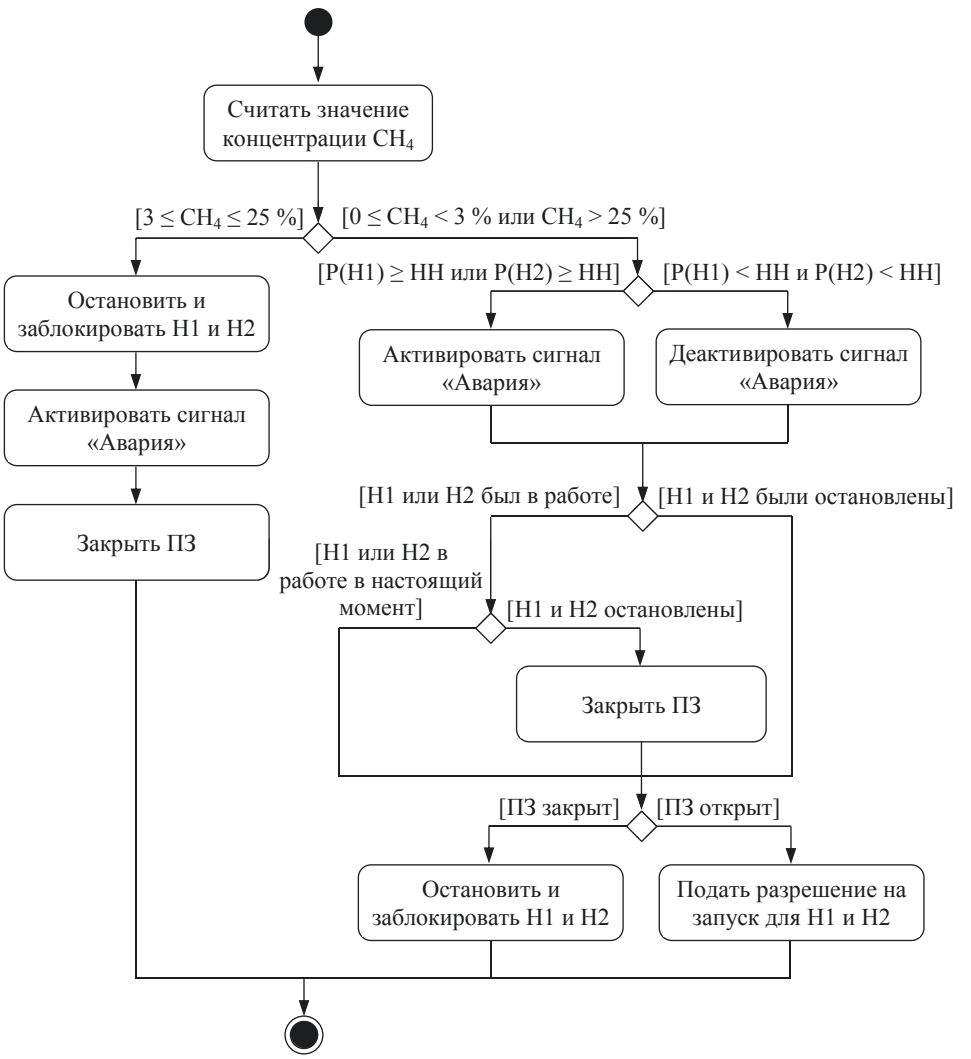

Рисунок 1. Требования к работе ПВНС в виде диаграммы видов деятельности
Figure 1. Requirements for an underground vacuum pumping station operation in the form of an activity diagram

регулирования и защиты. С тем, насколько качественно реализован технологический алгоритм, напрямую связана работоспособность всей системы автоматизации. Поэтому задача тестирования корректности реализации алгоритма является очень важной. Под корректностью в данном случае понимается соответствие программной реализации набору требований, которые предъявляются к системе (исходному алгоритму), и отсутствие в ней (реализации) ошибок.

Часто встречающийся на практике подход к тестированию ПО выглядит следующим образом: после формализации требований к ПО системы они (требования) передаются в отдел тестирования, в котором на их основе формируется список сценариев тестирования (тест-кейсов) — различных вариантов исходных данных и предполагаемого поведения системы. Этот список составляет тест-план, по которому тестировщик проверяет систему [11, 12].

Каждый тест-кейс описывается по определенному шаблону, который включает предусловия, набор действий для выполнения теста, проверку результатов работы и постусловия.

Составим список тест-кейсов для проверки требований к работе ПВНС (данный список не является полным).

Тест № 1. Запуск установки при нормальных условиях.

Тест № 2. Попадание значения концентрации CH4 в диапазон от 3 до 25 %.

Тест № 3. Попадание значения концентрации CH4 в диапазон от 0 до 3 % или выше 25 %.

Тест № 4. Возникновение давления на выходе насоса H1 больше аварийного порога либо равного ему.

Тест № 5. Давление на выходе насоса H1 меньше аварийного порога.

Тест № 6. Возникновение давления на выходе насоса H2 больше аварийного порога либо равного ему.

Тест № 7. Давление на выходе насоса H2 меньше аварийного порога.

Тест № 8. Один из насосов H1 или H2 был в работе, в данный момент оба насоса остановлены.

Тест № 9. Установка остановлена. Нажата кнопка «Закрыть ПЗ».

Тест № 10. Установка работает. Нажата кнопка «Закрыть ПЗ».

Тест № 11. Установка остановлена. Нажата кнопка «Открыть ПЗ».

Тест № 12. Установка работает. Нажата кнопка «Открыть ПЗ».

Тест № 13. На ПЗ отправлена команда «Закрыть», но он не закрылся.

Тест № 14. На ПЗ отправлена команда «Открыть», но он не открылся.

Пример развернутого описания теста.

Название: «Тест № 2. Попадание значения концентрации CH4 в диапазон от 3 до 25 %».

Предусловия:

– установка работает;

– концентрация CH4 равна 0 %;

– ПЗ открыт.

Действия:

– выставить значение CH4 в диапазоне от 3 до 25 %.

Реакция системы:

– DO1 (насос 1) — разомкнут;

– DO2 (насос 2) — разомкнут;

– DO3 («Авария») — замкнут;

– ПЗ закрыт.

Постусловия:

– выставить CH4 в значение, равное 0 %.

Основные недостатки ручного тестирования *заключаются в том, что:*

– процедура тестирования при большом количестве тест-кейсов может занимать значительное время;

– в процессе ручного тестирования из-за человеческого фактора могут возникать ошибки;

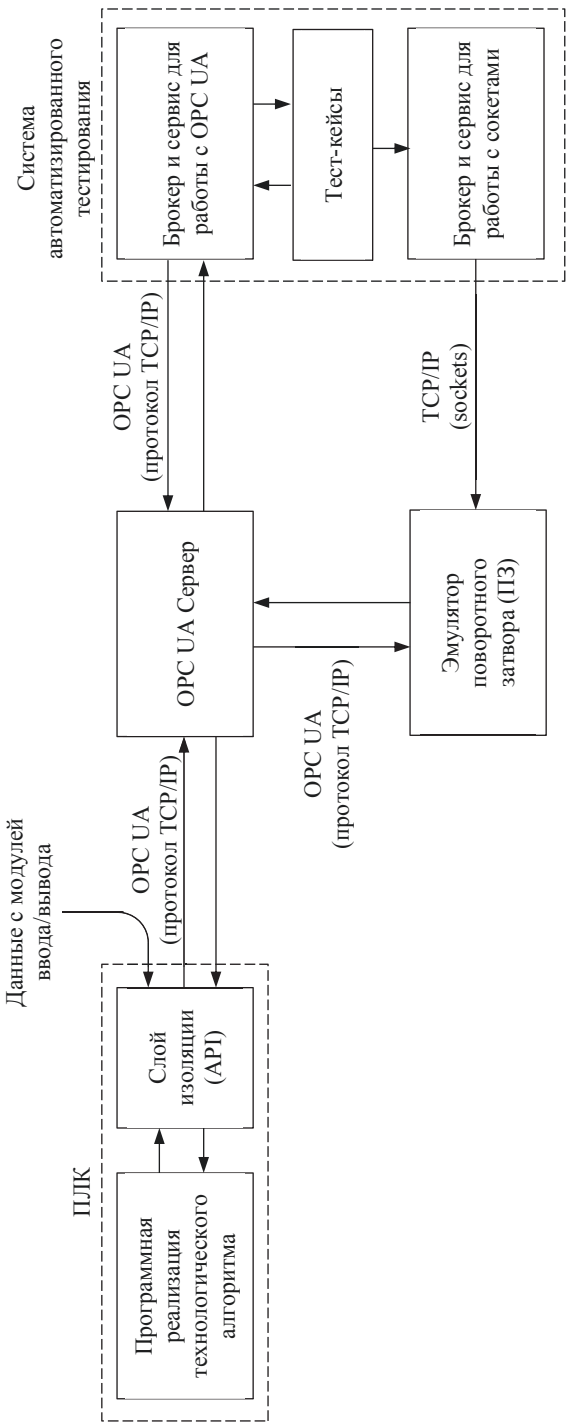

Рисунок 2. Система автоматизации тестирования технологического алгоритма ПВНС
Figure 2. Test automation system for an underground vacuum pumping station process algorithm

– для полноценного тестирования необходимо наличие тестового стенда.

Для повышения надежности и качества ПО необходимо автоматизировать процесс тестирования.

Автоматизация процесса тестирования технологического алгоритма.

Для автоматизации процесса тестирования необходимо:

– изолировать технологический алгоритм от системного слоя программы, что позволит при необходимости изменять источники и приемники данных без изменения самого технологического алгоритма;

– обеспечить возможность тестирования технологического алгоритма без аппаратного обеспечения;

– эмулировать работу управляемого технологического оборудования (поворотный затвор и т. д.);

– реализовать на доступном языке программирования: слой взаимодействия с технологическим алгоритмом, реализованным в рамках ПЛК; необходимый набор тест-кейсов.

Общий вид системы автоматизации тестирования технологического алгоритма ПВНС представлен на рис. 2.

Изоляция технологического алгоритма работы ПВНС. Основная идея изоляции технологического алгоритма работы ПВНС заключается в том, что доступ к контролируемым и управляемым параметрам системы должен осуществляться не напрямую (через связанные с входами/выходами ПЛК переменными), а посредством соответствующих функций.

Далее представлена часть программы, которая проверяет концентрацию метана в смеси и вырабатывает необходимые управляющие воздействия. При этом получение значения концентрации метана и управление производятся через переменные, связанные напрямую с входами/выходами ПЛК:

```
IF (CH4_AI >= 3 AND CH4_AI <= 25) THEN
    H1_DO := FALSE; // разомкнуть DO1 (насос 1)
    H2_DO := FALSE; // разомкнуть DO2 (насос 2)
    AVR_DO := TRUE; // замкнуть DO3 (сигнал «Авария»)
    IS_METH_OK := FALSE;
    CLOSE_ZVR();    // закрыть ПЗ
END_IF
```

Для создания слоя изоляции технологического алгоритма заменим переменные на функции:

```
IF (GET_CH4() >= 3 AND GET_CH4() <= 25) THEN
    SET_H1_STATE(FALSE);
    SET_H2_STATE(FALSE);
    SET_AVR_STATE(TRUE);
    IS_METH_OK := FALSE;
    CLOSE_ZVR();
END_IF
```

В этом случае вся работа с оборудованием и/или программными интерфейсами, через которые поступают и передаются данные, будет производиться в созданном наборе функций, что позволит при необходимости подменять источники и приемники данных без внесения изменений в технологический алгоритм.

Обеспечение возможности тестирования технологического алгоритма без аппаратного обеспечения. Некоторые средства разработки ПО для ПЛК, например такие как Codesys, позволяют запускать разработанные решения в среде, развернутой в операционной системе Windows, благодаря чему возможно какое-то время обходиться без аппаратного ПЛК. Если такой возможности нет, то посредством изоляции технологического алгоритма разработку можно вести с использованием ПЛК без набора модулей ввода/вывода, датчиков и других аппаратных элементов системы. В обоих этих случаях эмуляцию сигналов можно реализовать через интерфейс OPC UA [13, 14] либо какой-то иной способ связи ПЛК и персонального компьютера, который предоставляет производитель конкретного оборудования. Технологический алгоритм при этом не поменяется, изменится лишь содержание функций слоя изоляции (GET_CH4(), SET_H1_STATE() и т. д.). В этом случае данные будут считываться не с аппаратных модулей ввода/вывода, а через OPC UA теги, значения которых можно задавать через внешнее ПО.

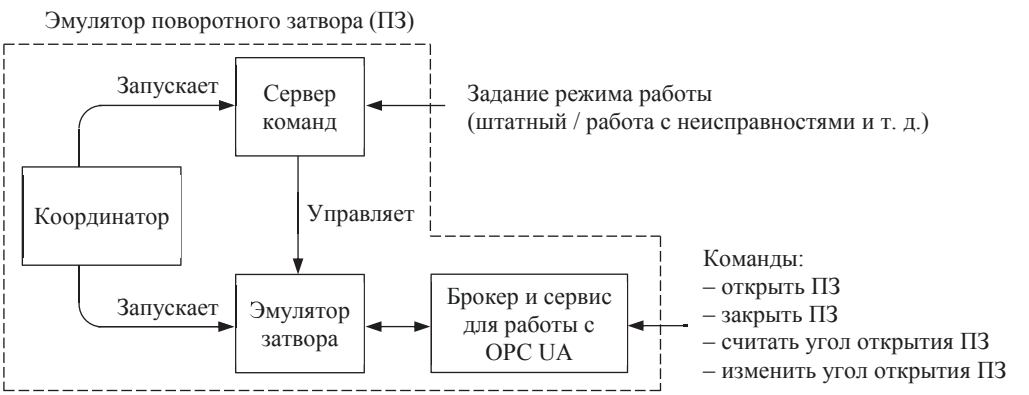

Рисунок 3. Структурная схема эмулятора ПЗ
Figure 3. Block diagram of a rotary gate emulator

Эмуляция работы управляемого технологического оборудования. Зачастую в состав системы автоматизации входит различное технологическое оборудование, которым необходимо управлять и контролировать его состояние. Это могут быть сложные устройства с нетривиальной логикой поведения. В системе автоматизации ПВНС таким устройством является поворотный затвор, управление которым должно осуществляться с ПЛК. В этом случае для выполнения полноценного автоматизированного тестирования реализации технологического алгоритма необходимо создать эмулятор, имитирующий работу такого устройства. Помимо штатной работы, эмулятор должен позволять имитировать различные нештатные ситуации, например: «был подан сигнал на ПЗ «закрыть», а он не закрылся». В реальности это может произойти по разным причинам: нарушение связи между ПЗ и ПЛК, отсутствие воздуха в пневмосистеме ПЗ (если это пневматический затвор) и т. д.

Работу эмулятора ПЗ с ПЛК проще всего реализовать через OPC UA интерфейс, посредством которого на эмулятор будут подаваться команды «открыть/закрыть ПЗ», а также будет считываться и изменяться текущий угол открытия затвора.

Система автоматизированного тестирования должна иметь возможность активации на эмуляторе ПЗ различных нештатных ситуации, которые были перечислены

ранее. Для имитации такого поведения лучше организовать отдельный канал для передачи соответствующих команд в эмулятор, например через сетевые сокеты, как правило, все современные языки программирования их поддерживают, либо воспользоваться каким-нибудь другим механизмом межпроцессного взаимодействия. Структурная схема эмулятора представлена на рис. 3.

Реализация слоя взаимодействия системы автоматизированного тестирования и технологического алгоритма на ПЛК и тест-кейсов для проверки корректности решения. Слой интеграции представляет собой набор программных модулей, представляющих интерфейс, и модели данных для взаимодействия функций программы, представляющих тест-кейсы с реализацией технологического алгоритма на ПЛК. В нашем случае было принята схема, представленная на рис. 2 – блок «Система автоматизированного тестирования». Брокеры – это программные компоненты, которые реализуют операции чтения/записи через интерфейс OPC UA и сетевые сокеты системы. На уровне брокеров не осуществляются проверки принимаемых и получаемых данных. Такого типа проверки реализуются на уровне базовых сервисов, которые осуществляют (при необходимости) структурную и логическую валидацию данных.

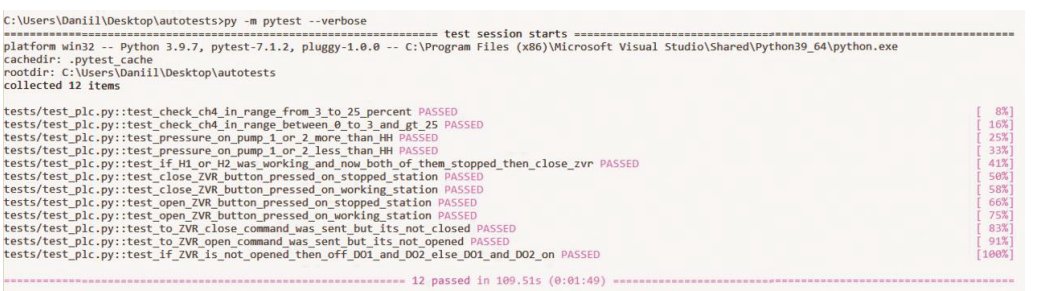

Рисунок 4. Вывод результатов тестирования
Figure 4. Test results output

Для реализации системы автоматизированного тестирования выбран язык программирования Python с учетом его простоты, достаточного количества необходимых библиотек и удобного каркаса тестирования (pytest) [15].

Программная реализация тест-кейса «Тест № 2. Попадание значения концентрации CH4 в диапазон от 3 до 25 %» имеет следующий вид:

```
  def test_check_ch4_in_range_from_3_to_25_percent():
    """Тест 2: Попадание значения концентрации CH4 в диапазон
от 3 до 25 %"""
    # Предусловия
    prep_state() # установка значений контролируемых параметров
на значения "по умолчанию"

    # Запуск установки (если была остановлена)
    # Эмулируем концентрацию CH4 = 1 %
    opcua_service.set_tag_value(ns=4, tag_name=AI_CH4_tag,
value=1.0, tp=ua.VariantType.Float)
    # Открываем затвор (эмуляция нажатия кнопки Открыть затвор)
    open_zvr_by_button()
```

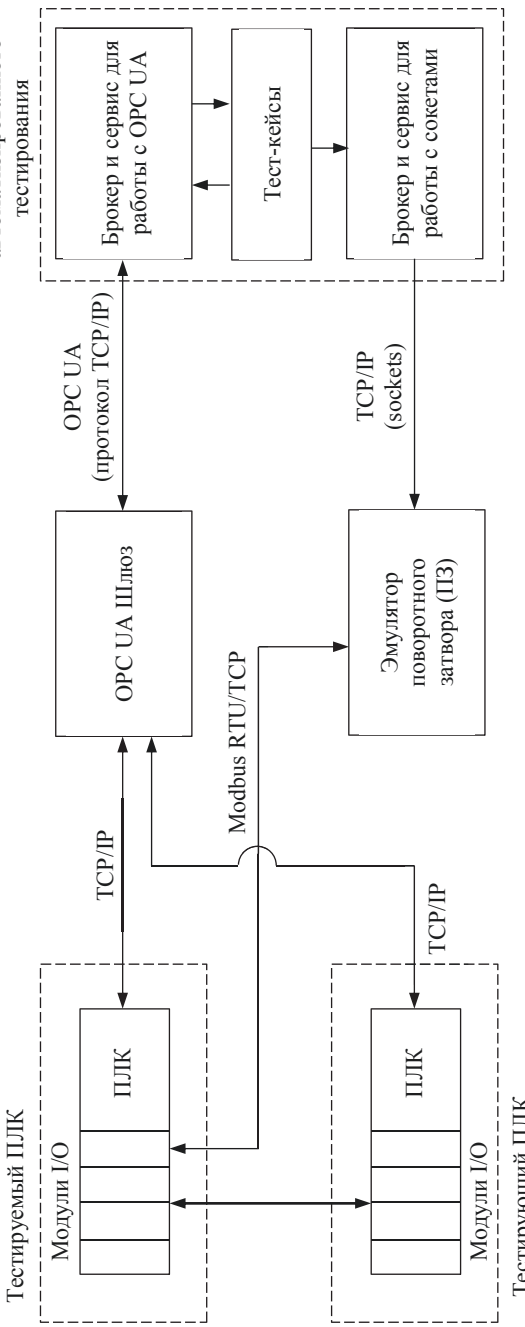

Рисунок 5. Структура системы для проведения интеграционного тестирования
Figure 5. Structure of the integration testing system

```
    # Действия
    opcua_service.set_tag_value(ns=4, tag_name=AI_CH4_tag,
value=3.0, tp=ua.VariantType.Float)
    time.sleep(zvr_wait)  # Ждем пока закроется затвор

    # Проверка реакции системы
    zvr_angle = opcua_service.get_tag_value(ns=4, tag_name=zvr_
angle_tag)
    DO_H1_value = opcua_service.get_tag_value(ns=4, tag_
name=DO_H1_tag)
    DO_H2_value = opcua_service.get_tag_value(ns=4, tag_
name=DO_H2_tag)
    DO_AVR_value = opcua_service.get_tag_value(ns=4, tag_
name=DO_AVR_tag)

    assert zvr_angle <= 5
    assert DO_H1_value is False
    assert DO_H2_value is False
    assert DO_AVR_value is True

    # Постусловия
    # Эмулируем концентрацию CH4 = 1 %
    opcua_service.set_tag_value(ns=4, tag_name=AI_CH4_tag,
value=1.0, tp=ua.VariantType.Float)
```

Как можно видеть, приведенный программный код близок по своей структуре и содержанию словесному описанию тест-кейса, что значительно упрощает разработку такого типа программ. Остальные тест-кейсы были реализованы похожим образом.

Для запуска процесса тестирования используется программный модуль pytest. Если все тесты проходят проверку, то результат будет иметь вид, представленный на рис. 4, в примере было реализовано 12 тестов, все они завершились успешно.

Анализ возможности применения предлагаемого подхода в рамках других групп тестирования. В статье рассматриваются построение инфраструктуры и методика ее применения для проведения юнит-тестирования, которое, фактически, является первым уровнем группы функциональных тестов. Предлагаемый подход ограниченно применим для интеграционных тестов. Интеграционное тестирование предполагает проверку связи между компонентами системы и корректность обмена информацией между ними. Система автоматизации ВНС включает в себя ПЛК, модули ввода/вывода, датчики, исполнительные механизмы и оборудование для обеспечения связи ПЛК и SCADA-уровня. Структура системы для проведения интеграционного тестирования представлена на рис. 5. Тестирующий ПЛК – это ПЛК, через который система тестирования оказывает воздействие на модули ввода тестируемого ПЛК (через модули вывода) и осуществляет контроль срабатывания релейных выходов через модули ввода. Эмулятор поворотного затвора потребуется доработать в части возможности работы по протоколу Modbus RTU/TCP.

В данном случае не предусматривается проверка с подключенными датчиками и исполнительными механизмами, так как это потребовало бы помещения их в специальную среду, состоянием которой нужно было бы управлять, что, на наш взгляд, кажется излишним. В остальном связь и корректность передачи

информации может быть проверена. Приемочное тестирование такого рода систем чаще всего проводится в ручном режиме с полным комплектом подключенного оборудования, в этом случае автоматизированное тестирование не применяется, но можно допустить, что на первом этапе приемочного тестирования может быть проверена корректность реализации алгоритма через запуск интеграционных тестов.

Что касается группы нефункциональных видов тестирования, с использованием предлагаемой инфраструктуры можно проводить тестирование надежности работы системы, для этого потребуется стенд интеграционного тестирования и модификация в компоненте «Система автоматизированного тестирования» (рис. 5), которая позволила бы запускать тесты не однократно, а в циклическом режиме.

Предлагаемый подход позволит довольно эффективно проводить тестирование, связанное с изменениями, а именно: дымовое тестирование и регрессионное тестирование. Для дымового тестирования можно выделить набор тестов, прохождение которых будет говорить об общей работоспособности и корректности алгоритма. При этом время прохождения дымового тестирования значительно меньше, чем время, которое требуется для выполнения всех тестов. Выявленные в процессе отладки и тестирования программы ошибки могут стать источником дополнительного набора тестов, которые не были учтены в процессе составления тест-плана по техническому заданию. Они будут являться базой для проведения регрессионного тестирования.

Выводы. Приведенный в статье подход к автоматизации тестирования программной реализации технологического алгоритма работы ПВНС можно применять для тестирования систем автоматизации различных технологических процессов на базе ПЛК. Это позволит значительно сократить время на ручное тестирование, уменьшить количество ошибок в программе, оперативно вносить изменения в ПО в процессе наладки и ввода системы в эксплуатацию. Данный подход позволит начать работу над реализацией технологического алгоритма с поддержкой тестирования до непосредственного развертывания ПО на аппаратном обеспечении (ПЛК). Эмуляторы работы технологического оборудования являются переиспользуемыми компонентами и, в случае необходимости, могут применяться в работе над различными проектами, которые разрабатываются в рамках предлагаемого подхода.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Рудничная вентиляция / под ред. Ушакова. 2-е изд., перераб. и доп. М.: Недра, 1988. 440 с.

2. Карпов Е. Ф., Басовский Б. И. Контроль проветривания и дегазации в угольных шахтах. М.: Недра, 1994. 336 с.

3. Шевченко Л. А. Расчет параметров глубокой дегазации угольных шахт // Известия вузов. Горный журнал. 2011. № 5. С. 45–49.

4. Карпов Е. Ф., Рязанов А. В. Автоматизация и контроль дегазационных систем. М.: Недра, 1983. 190 с.

5. Малашкина В. А., Вострикова Н. А. Анализ основных особенностей транспортирования метановоздушной смеси по подземному вакуумному дегазационному трубопроводу // ГИАБ. 2004. № 8. С. 253–257.

6. Кузнецов В. И. Механические вакуумные насосы. М.: Госэнергоиздат, 1983. 279 с.

7. Малашкина В. А. Дегазационные установки. М.: МГГУ, 2007. 189 с.

8. Gary L. Pratt PE. The book of CODESYS: the ultimate guide to PLC and industrial controls programming with the CODESYS IDE and IEC 61131-3. ControlSphere LLC., 2021. 492 p.

9. Suriya Sundaramoorthy. UML diagramming: a case study approach. Auerbach Publications, 2022. 402 p.

10. Alan Dennis, Barbara Wixom, David Tegarden. Systems analysis and design: an object-oriented approach with UML. Wiley, 2020. 544 p.

11. Kristin Jackvony. The complete software tester: concepts, skills, and strategies for high-quality testing. Independently published, 2022. 512 p.
12. James D. McCaffrey. Software testing: fundamental principles and essential knowledge. Booksurge Publishing, 2009. 120 p.
13. John Rinaldi. OPC UA – unified architecture: the everyman's guide to the most important information technology in industrial automation. CreateSpace Independent Publishing Platform., 2016. 170 p.
14. John Rinaldi. OPC UA: the Basics: an OPC UA overview for those who may not have a degree in embedded programming paperback. CreateSpace Independent Publishing Platform, 2013. 76 p.
15. Brian Okken. Python testing with pytest: simple, rapid, effective, and scalable. Pragmatic Bookshelf, 2022. 274 p.

Поступила в редакцию 10 ноября 2022 года

Сведения об авторах:

Абдрахманов Марат Ильдусович – кандидат технических наук, доцент кафедры автоматики и компьютерных технологий Уральского государственного горного университета. E-mail: marat-ab@mail.ru; https://orcid.org/0000-0002-0391-6204
Гардт Даниил Алексеевич – магистрант кафедры информатики Уральского государственного горного университета. E-mail: 2g0d0a0@gmail.com

DOI: 10.21440/0536-1028-2023-3-113-126

Developing a unit testing framework for control system software implementation of a degassing vacuum pumping station

Marat I. Abdrakhmanov[1], Daniil A. Gardt[1]
[1] Ural State Mining University, Ekaterinburg, Russia.

Abstract

Research objective *is to develop a test automation system for the process algorithm software implementation of the degassing vacuum pumping station control system operation at a coal mine.*
Relevance. *The vacuum pumping station is an essential component of a mine degassing system. Degassing system operation depends on its correct functioning. A programmable logic controller with appropriate software is a key component of the vacuum pumping station automation system. Correct software implementation of the system operation process algorithm is of primary importance. To verify the compliance of the implementation with the requirements, manual testing is commonly used according to a pre-compiled test-plan, which is formed based on the requirements imposed on the system. Manual testing has a number of disadvantages. It is a long process where human errors are likely to occur, and the use of a test bench and/or an assembled control system sample is inevitable. Requirements for the algorithm can be changed in the course of program development, debugging, and initiation, which greatly complicates the testing process.*
Results. *To solve the problem of process algorithm software implementation testing automation, the process control algorithm should be isolated from the software components responsible for the hardware. Access to the process control algorithm should be provided through standard interfaces (for example, OPC UA). Additional process equipment (valves, etc.) used in the automation system should be implemented in the form of emulators with an opportunity to assign their modes of operation. By creating an additional software layer of interaction with a process algorithm and emulator in a programming language with advanced test tools (for example, Python), it is possible to implement test scenarios from a test plan in this language. It will allow testing in automatic mode rather than manually.*
Conclusions. *The proposed approach to process algorithm software implementation testing automation will greatly simplify software development for this type of automation and control systems.*

Keywords: *degassing; vacuum pumping station; automation; testing; manual testing; automated testing.*

REFERENCES

1. Ushakov K. Z. (ed.) *Mine ventilation. 2nd edition, revised and enlarged*. Moscow: Nedra Publishing; 1988. (In Russ.)

2. Karpov E. F., Basovskii B. I. *Control of ventilation and degassing in coal mines*. Moscow: Nedra Publishing; 1994. (In Russ.)

3. Shevchenko L. A. Calculation of the parameters of deep de-gassing of coal. *Izvestiya vysshikh uchebnykh zavedenii. Gornyi zhurnal = News of the Higher Institutions. Mining Journal*. 2011; 5: 45–49. (In Russ.)

4. Karpov E. F., Riazanov A. V. *Automation and control of degassing systems*. Moscow: Nedra Publishing; 1983. (In Russ.)

5. Malashkina V. A., Vostrikova N. A. Analyzing the main features of methane-air mixture transportation through an underground vacuum degassing pipeline. *Gornyi informatsionno-analiticheskii biulleten (nauchno-tekhnicheskii zhurnal) = Mining Informational and Analytical Bulletin (scientific and technical journal)*. 2004; 8: 253–257. (In Russ.)

6. Kuznetsov V. I. *Mechanical vacuum pumps*. Moscow: Gosenergoizdat; 1983. (In Russ.)

7. Malashkina V. A. *Degassing units*. Moscow: MSMU Publishing; 2007. (In Russ.)

8. Gary L. Pratt PE. *The book of CODESYS: the ultimate guide to PLC and industrial controls programming with the CODESYS IDE and IEC 61131-3*. ControlSphere LLC., 2021.

9. Suriya Sundaramoorthy. *UML diagramming: a case study approach*. Auerbach Publications; 2022.

10. Alan Dennis, Barbara Wixom, David Tegarden. *Systems analysis and design: an object-oriented approach with UML*. Wiley, 2020.

11. Kristin Jackvony. *The complete software tester: concepts, skills, and strategies for high-quality testing*. Independently published; 2022.

12. James D. McCaffrey. *Software testing: fundamental principles and essential knowledge*. Booksurge Publishing; 2009.

13. John Rinaldi. *OPC UA – unified architecture: the everyman's guide to the most important information technology in industrial automation*. CreateSpace Independent Publishing Platform; 2016.

14. John Rinaldi. *OPC UA: the Basics: an OPC UA overview for those who may not have a degree in embedded programming paperback*. CreateSpace Independent Publishing Platform; 2013.

15. Brian Okken. *Python testing with pytest: simple, rapid, effective, and scalable*. Pragmatic Bookshelf; 2022.

Received 10 November 2022

Information about the authors:

Marat I. Abdrakhmanov – PhD (Engineering), associate professor of the Department of Automation and Computer Technology, Ural State Mining University. E-mail: marat-ab@mail.ru; https://orcid.org/0000-0002-0391-6204

Daniil A. Gardt – master's student, Department of Informatics, Ural State Mining University. E-mail: 2g0d0a0@gmail.com

Для цитирования: Абдрахманов М. И., Гардт Д. А. Разработка инфраструктуры для проведения юнит-тестирования программной реализации системы управления дегазационной вакуум-насосной станцией // Известия вузов. Горный журнал. 2023. № 3. С. 113–126. DOI: 10.21440/0536-1028-2023-3-113-126

For citation: Abdrakhmanov M. I., Gardt D. A. Developing a unit testing framework for control system software implementation of a degassing vacuum pumping station. *Izvestiya vysshikh uchebnykh zavedenii. Gornyi zhurnal = Minerals and Mining Engineering*. 2023; 3: 113–126 (In Russ.). DOI: 10.21440/0536-1028-2023-3-113-126