

**ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРОЕКТИРОВАНИЯ
ГОРНО-ТЕХНИЧЕСКИХ СИСТЕМ
THEORETICAL FOUNDATIONS OF THE DESIGN OF MINING SYSTEMS**

Научная статья

УДК 622.831

DOI: 10.26730/1999-4125-2025-6-104-113

**МЕТОДИКА РАЗРАБОТКИ ПРОГРАММНОГО КОМПЛЕКСА ЭМУЛЯЦИИ
СИСТЕМ ДИСПЕТЧЕРСКОГО УПРАВЛЕНИЯ И СБОРА ДАННЫХ (SCADA)**

Липина Галина Александровна¹,
Липин Артем Вадимович²

¹ Кузбасский государственный технический университет имени Т. Ф. Горбачева² ООО «Сомерсет Интернэшнл Раша»

* для корреспонденции: galipina@yandex.ru

**Информация о статье**

Поступила:

01 октября 2025 г.

Одобрена после

рецензирования:

15 ноября 2025 г.

Принята к публикации:

02 декабря 2025 г.

Опубликована:

22 декабря 2025 г.

Ключевые слова:

эмуляция, программный комплекс, производственный процесс, обучение персонала, математическая модель, обогащение полезных ископаемых, угольная отрасль, флотация

Аннотация.

В данной статье рассматривается актуальность и методика создания программного комплекса эмуляции систем диспетчерского управления и сбора данных (SCADA) в целях обучения производственного диспетчерского персонала как одно из направлений в сфере цифровизации производства для повышения эффективности технологического процесса, а также снижения сопутствующих издержек вследствие уменьшения количества аварийных ситуаций и общего времени простоев. В статье на примере создания одного из элементов (классов) программного комплекса, а именно эмулятора флотомашины как широко распространенного вида горно-обоганительного оборудования описывается поэтапный процесс реализации данной структурной единицы, и далее общая концепция работы системы. При описании подхода к созданию класса выделяются несколько основных этапов, а именно этапы определения основных операционных переменных, системных переменных, законов их взаимодействия, обобщение физических процессов, протекающих при функционировании технологического оборудования для получения достоверной картины его работы при упрощенном математическом аппарате. Несмотря на то, что в данной статье рассматривается лишь общий подход к получению программного комплекса, без привязки к определенному языку программирования, она опирается на процесс реализации существующего программного продукта и обобщает опыт, ранее полученный при создании работоспособной модели. В качестве сопроводительных иллюстраций в статье приведены скриншоты готового программного продукта. Однако реализация данной идеи может существенно различаться, как и уровень абстрактизации при создании классов, и зависит прежде всего от целей и задач, стоящих перед разработчиком.

Для цитирования: Липина Г.А., Липин А.В. Методика разработки программного комплекса эмуляции систем диспетчерского управления и сбора данных (SCADA) // Вестник Кузбасского государственного технического университета. 2025. № 6 (172). С. 104-113. DOI: 10.26730/1999-4125-2025-6-104-113, EDN: KFKZAK

Введение (Introduction). В настоящее время все большую актуальность приобретают направления рационализации производственных процессов, направленные на снижение издержек и повышение эффективности и безопасности труда. Данные процессы затрагивают практически все сферы народного хозяйства и этапы производства. Повышение общей эффективности предприятия – главная задача руководителей и инженеров, которая включает в себя множество направлений, от технических – таких как улучшение технологии процесса, повышения уровня автоматизации и обновления парка машин и механизмов, до организационных, включающих в себя повышение квалификации персонала и уровня мотивации персонала, улучшение условий труда и снижение уровня производственного травматизма [1].

Одним из репрезентативных критериев эффективности современного производства является уровень его цифровизации [2]. Централизованное управление и высокий уровень автоматизации позволяют не только эффективно управлять и контролировать уже отлаженные процессы, но и анализировать большой объем данных для своевременного выявления слабых и неэффективных этапов. Это позволяет как значительно сократить сроки отладки вновь запущенных производств, так и непрерывно улучшать уже существующие, за счет существенного облегчения доступа к информации для профильных специалистов и технологов [3].

Однако на этом возможности цифровизации производства не исчерпываются. Наряду с производственными и бизнес-процессами огромное значение имеют процессы обучения и повышения квалификации персонала. Роль, которую играет личная выучка рядового, и, в особенности, диспетчерского персонала, сложно переоценить. Действительно, ведь с повышением уровня автоматизации производства большой объем оперативной информации замыкается на узком круге лиц, в зоне ответственности которых могут находиться огромные мощности, причем даже незначительная их ошибка в оценке входных данных либо неверные действия в ходе возникновения внештатной ситуации могут поставить крест на эффективности всех остальных процессов, приводя как к серьезным простоям и недовыпуску продукции, так и к тяжелейшим авариям и несчастным случаям, ведущим к долговременным тяжелым последствиям для всего предприятия в целом [4]. Это настоящее узкое место современного производства, когда высочайший уровень ответственности сталкивается с недостаточной квалификацией и низким уровнем мотивации, помноженным на человеческий фактор. К

сожалению, в настоящее время даже самая продуманная система автоматических блокировок и взаимодействий не может полностью заменить человека, а лишь выступает в качестве подстраховки. Надо учитывать и то, что зачастую даже на опасных производственных объектах логику работы автоматизации производства и системы безопасности реализуют не профильные специалисты, а промышленные программисты подрядных организаций, пусть и имеющие высочайшую квалификацию в своей области, но далекие от специфики конкретного производства. А техническое задание для них пишут сотрудники и технологи с предприятия-заказчика, производственники, зачастую вообще не знакомые с программированием. Эта система рождает огромное количество коллизий, которое мы называем «детскими болезнями» и которое тяжким бременем ложится на диспетчеров и операторов, что в случае их недостаточной квалификации может привести к тяжелым последствиям.

К сожалению, самым эффективным и действенным способом подготовки диспетчерского персонала является практика. Даже при условии наличия крепкой теоретической базы только по прошествии определенного количества времени и приобретения опыта, связанного с самостоятельным решением конкретных специфических задач, диспетчер становится эффективным звеном производственного процесса, повышая, а не понижая его эффективность, а также важной и надежной частью системы безопасности предприятий [5]. Таким образом, именно снижение длительности периода адаптации и обучения диспетчерского персонала является одной из важных задач, стоящих перед ИТР предприятия. И несмотря на то, что как на производствах, так и сторонними организациями созданы обучающие центры, а в правилах промышленной безопасности прописаны правила стажировок и дублирования, данная задача остается актуальной [6].

Методы (Methods). Одним из методов решения задачи практической тренировки диспетчерского персонала является создание реалистичных тренажеров, адаптированных под конкретное производство и позволяющих полностью эмулировать как производственный процесс, так и саму программную среду, вплоть до элементов графического дизайна, что также является важной деталью обучения [7]. Ведь огромное значение, особенно в стрессовой ситуации, имеет скорость реакции диспетчера, причем его действия при привычных элементах графического окружения принимают рефлекторный характер. Также система эмуляции не должна быть зашифрованной и

позволять обучаемому с легкостью определять закономерности своей работы, провоцируя его на изучение самого тренажера и заучивание его особенностей, а не на обучение тонкостям управления реальным производственным процессом. В состав программного комплекса должен включаться модуль оценки уровня подготовки обучаемого, причем сама оценка должна максимально отражать подготовленность специалиста к реалиям производства. Например, если перед работником обогатительной фабрики стоит задача обеспечить за определенный отрезок времени максимальное количество угля на конвейере при условии недопущения аварийных ситуаций и с минимальным расходом электроэнергии и химических реагентов для флотации, а также соответствие угля заданным характеристикам по зольности и влажности именно эти критерии и должны оцениваться программой, выдавая определенную усредненную оценку [8-9].

Все вышеперечисленное формирует определенные требования к программному комплексу эмуляции производственного процесса в целях обучения диспетчерского персонала, далее по тексту – эмулятор производственного процесса (ЭПП).

1. Модульность структуры, именно создание структурных единиц – классов [10], соответствующих различным технологическим единицам оборудования, с набором параметров и входных/выходных переменных [11]. Далее в качестве примера будет рассмотрен класс «Флотомашина» (ФМ).

2. Унификация построения структуры из классов за счет транзитных кросс-классов, таких как «Трубопровод».

3. Обеспечение единого графического стиля, основанного на государственных стандартах (ГОСТ) условно графических изображений (УГО) с возможностью адаптации под индивидуальные особенности графического стиля SCADA-систем отдельных предприятий.

4. Наличие механизма настраиваемой генерации случайных аварий из пула аварий с возможностью их ликвидации при совершении корректных действий со стороны оператора, в том числе в результате ответа на всплывающие текстовые вопросы/события.

5. Модуль оценки эффективности действий оператора.

При разработке программного комплекса, особенно на подготовительных этапах, следует активно применять научные методы обобщения, а именно метод анализа и синтеза [12]. Мысленное разложение объекта на составные части для выявления общего и особенного, с дальнейшим объединением отдельных элементов в единое целое позволяет добиться понимания сущности объекта и серьезно упростить как

математическую модель, так и количество необходимого кода.

Результаты исследования (Results): В качестве примера, а также результата исследования рассмотрим создание программного класса флотомашины (ФМ). Данный класс наглядно иллюстрирует принцип создания классов, а сам процесс флотации в целом и ФМ в частности является структурно-образующим звеном технологической цепи множества обогатительных фабрик.

Для начала стоит отметить, что в данной статье не будет приведен исходный код класса, так как реализация целиком и полностью зависит от программной среды и навыков программиста и не привязана к какому-либо языку программирования.

Так что же представляет из себя ФМ? В обобщенном виде это система разделения веществ и фракций, основанная на эффектах смачиваемости, плавучести и электростатических взаимодействий. Проще говоря – система создания определенных условий, при которых нужно всплывает, а ненужное тонет. Несомненно, при создании математических моделей эффективности обогащения, формулы принимают весьма громоздкие формы, и зачастую основаны на существенном количестве статистических данных, полученных практическим путем, наложенных на закон нормального распределения. Однако для реализации наших целей достаточно описания основных физических процессов и взаимодействий, крупными мазками моделирующих изменения характеризующих величин и переменных, для получения определенного уровня достоверности без применения заскриптованных сценариев и «хардкода». Например, при изменении уровня столба жидкости за счет гидростатического давления расход жидкости в трубном отводе в нижней части ФМ увеличивается. Данный процесс описывается парой формул гидростатического давления и расхода жидкости через проходное сечение. Это придаст модели перетекания жидкости достоверности, однако не учтет ряда процессов, непосредственно связанных с технологией, например, осаждения. С целью приближения к полноценному моделированию производственного процесса хорошей идеей будет внесение ряда коэффициентов и взаимосвязей, позволяющих повысить уровень достоверности моделирования самого процесса обогащения на основе практических данных, полученных непосредственно на производстве. Например, зависимость плотности флотоконцентрата от количества добавляемых реагентов, работы импеллеров ФМ, транспортирующих воздух в процесс, пересчет плотности хвостов флотации в

соответствии с плотностью и объемом флотоконцентрата, который в свою очередь зависит от уровня в ФМ, уровня перелива, и в том числе от работы пеногонов.

В самом деле описание каждого класса, от «насоса» до «вакуумного пресс-фильтра» требует детального понимания работы этого самого оборудования и заслуживает отдельного рассмотрения, поэтому в данной статье мы рассмотрим лишь общий принцип. Уже сейчас читателю становится понятно, что главным залогом успеха решения подобной задачи становится тесная связка работы программиста и опытного производителя-технолога либо квалифицированного аппаратчика. Однако в целях структуризации процесса допустимо выделить общий «алгоритм» создания класса, разделенный на несколько основных этапов.

1. Этап упрощения и абстракции. В ходе этого этапа мы максимально упрощаем механизм, представляя его в виде некоего черного ящика, который имеет входные и выходные величины. Для флотомшины входными величинами (в контексте нашей задачи) являются:

- входной расход пульпы $W_{вх}$ [м³/ч];
- входная плотность (содержание

твердого) $D_{вх}$ [г/л].

Выходные величины:

- расход хвостов флотации $W_{хв}$ [м³/ч];
- расход флотоконцентрата $W_{фк}$ [м³/ч];
- плотность флотоконцентрата $D_{фк}$ [г/л].

2. Этап выделения переменных управления, функциональных переменных, настроек и коэффициентов. В ходе этого этапа мы определяем переменные и коэффициенты, необходимые для работы нашего класса. Например, переменными управления для флотомшины являются:

- Уровень заданный L^* [%];
- расход реагента (вспениватель) $W1$ [кг/т];
- расход реагента (собиратель) $W2$ [кг/т].

Функциональные (системные) переменные:

- текущий уровень L_c [%];
- площадь сечения S (открытия) флото-клапана(ов) [%];
- Количество импеллеров в работе [шт];
- Количество пеногонов в работе [шт].

Коэффициенты взаимосвязей:

- коэффициент влияния вспенивателя $K1$;
- коэффициент влияния собирателя $K2$;
- коэффициент перелива (связь расхода флотоконцентрата и разницы текущего уровня, и

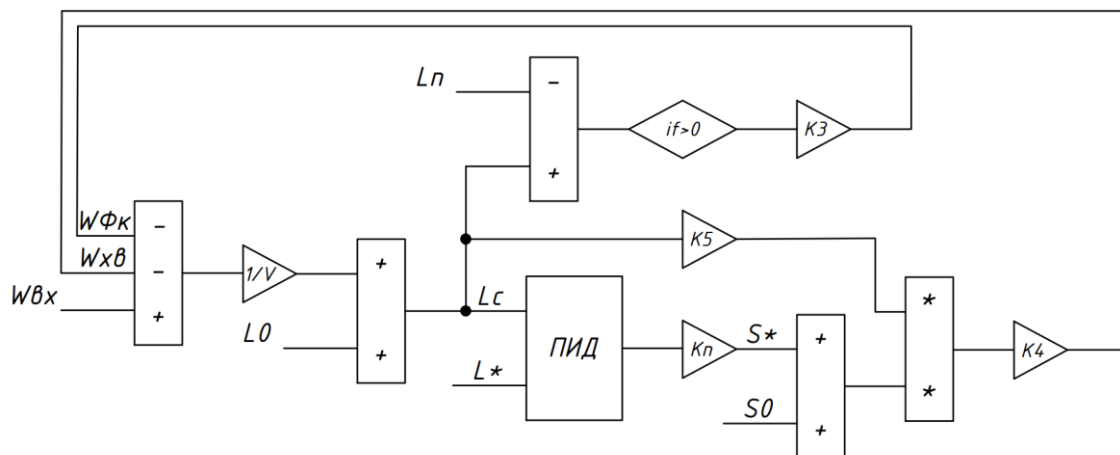


Рис. 1. Упрощенная модель функциональной схемы ФМ
Fig. 1. Simplified model of the functional diagram of the FM

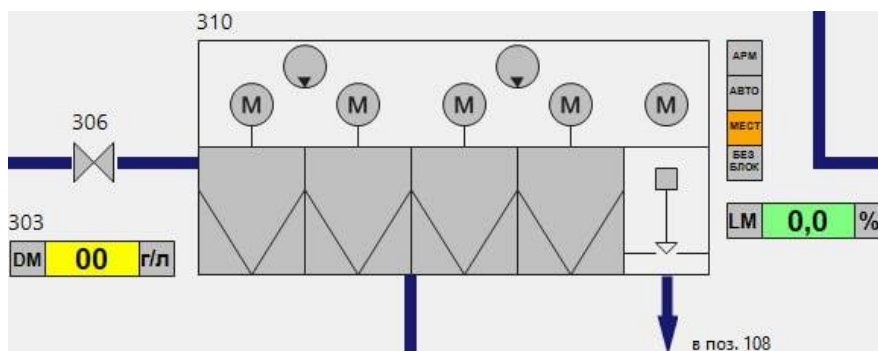


Рис. 2. Графическое отображение ФМ на SCADA-мнемосхеме
Fig. 2. FM graphical representation on a SCADA mimic diagram

уровня перелива) К3;

- коэффициент расхода хвостов, связь расхода хвостов флотации с сечением клапанов флотомашины К4;

- коэффициент расхода хвостов, связь расхода хвостов флотации с текущим уровнем флотомашины К5;

- коэффициенты ПИД-регулятора уровня флотомашины.

Сервисные (настроечные) переменные:

- объем флотомашины V [м³];
- наличие пеногонов;
- уровень перелива Lп [%];
- верхняя предупредительная граница уровня (ВПП) [%];
- нижняя предупредительная граница уровня (НПП) [%];
- верхняя аварийная граница уровня (ВАГ) [%];
- нижняя аварийная граница уровня (НАГ) [%].

3. Этап определения структурных схем. На данном этапе следует определить основной принцип взаимодействия переменных, на основе которых будет строиться математическая модель. Например, упрощенная структурная схема [6] взаимодействия потоков и уровней для ФМ представлена на Рис. 1.

4. Этап формирования УГО класса и интерфейса пользователя. УГО должен не только соответствовать принятым в отрасли стандартам, но и иметь возможность оптимизации под внутренние (исторически сложившиеся) стандарты конкретного предприятия, которые в некоторых случаях могут несколько разниться. Для примера, на Рис. 2 приведен вариант ФМ, реализованной в ЭПП, а на Рис. 3 приведен

вариант интерфейса управления.

5. Этап реализации. На данном этапе окончательно формируется математическая модель и реализуется в виде программного кода.

Для регулятора уровня флотомашины применяется классический ПИД регулятор [13]:

$$L(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de}{dt}; \quad (1)$$

где K_p, K_i, K_d – коэффициенты усиления пропорциональной, интегрирующей и дифференцирующей составляющих регулятора [5].

Коэффициенты усиления ПИД регулятора допускается настраивать во время отладки, при «запуске» системы, и более того, данный метод является наиболее близким к реальностям ПНР на производстве [14].

Учитывая особенности математического описания технологических процессов в обогащении, при моделировании рекомендуется применять фаззи-логику, позволяющую в полной мере реализовать табличную структуру зависимостей, полученных практическим путем, а также системы машинного обучения [15].

6. Этап детализации и аварийности. На данном, финальном этапе необходимо продумать типы и законы возникновения неисправностей [16], возникающих во время работы реализуемого объекта, как во время его нормальной работы, так и в принудительном порядке. Неисправности можно разделить на два вида – возникающие в результате ошибочных действий оператора и принудительные, которые не зависят от действий оператора и формируются системой, отражая действительность реального производства.

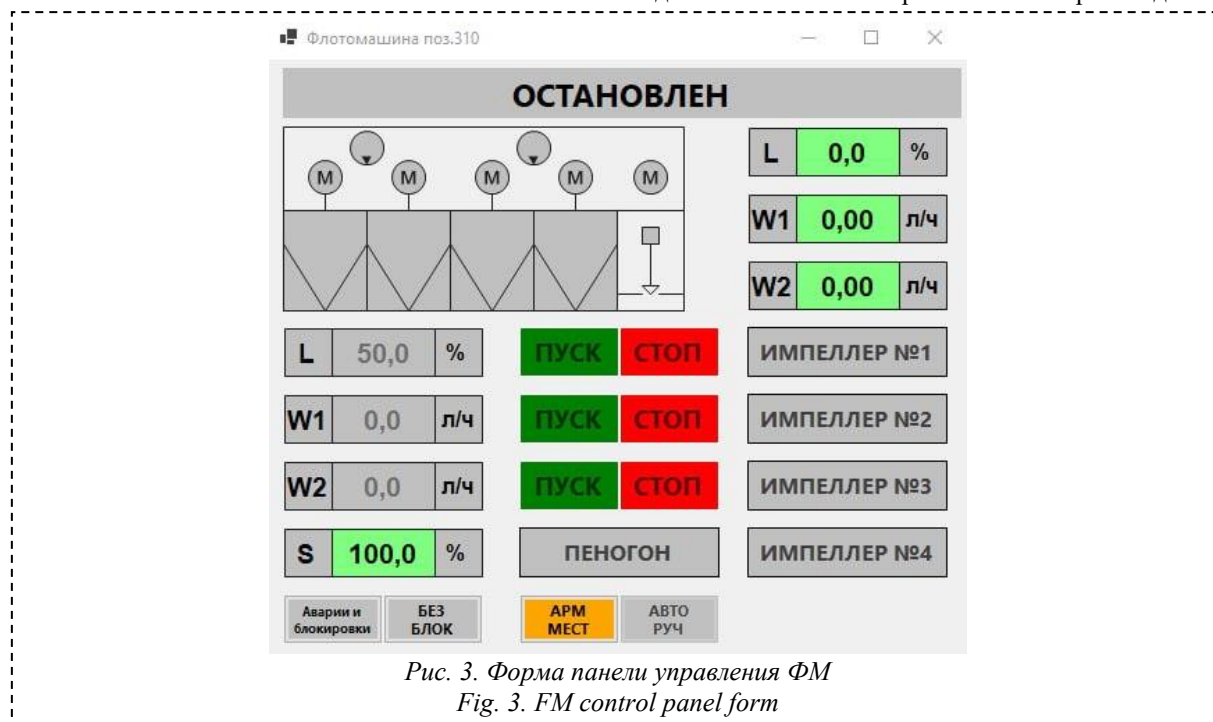


Рис. 3. Форма панели управления ФМ
Fig. 3. FM control panel form

Однако сам класс не должен разделять эти неисправности, механизм их выявления универсален и соответствует возможностям технических средств автоматизации. Например, если на механизме из датчиков установлен лишь датчик уровня, то единственная возможность системы сообщить оператору об аварийной ситуации – через показания датчика уровня. Система не может выдать ошибку, скажем, «перегрев двигателя», если комплектация двигателя не предусматривает наличия датчиков температуры или термосопротивлений. Если аварийное событие моделирует неисправность самого датчика уровня, причем оно связано не с терминальным разрушением, которое вполне может быть зафиксировано САУ, а, скажем, с заиливанием патрубка датчика дифференциального давления, оператор должен сам догадаться о неисправности по косвенным показателям – таким как несоответствие отображаемого уровня текущему режиму работы. Типовые аварии ФМ приведены на Рис. 4.

При превышении установленных НАГ и ВАГ уровня либо несоответствии заданных и текущих показаний уровня, система формирует соответствующие аварии независимо от причин,

их вызвавших. Однако для их устранения требуются различные условия. Например, если уровень превышен в результате слишком большого задания для насоса питания флотации и поток превышает пропускные возможности клапанов флотомашины, достаточно скорректировать поток и сбросить аварию. Но если аварию сгенерировала система, то этого недостаточно. Система устанавливает триггер, соответствующий предустановленной аварии, например, «заклинивание клапана ФМ». Устранить аварию путем «сброса» невозможно, при попытке вновь запустить ФМ авария повторится. В данном случае оператору необходимо решить тестовое задание в виде выбора вариантов действий во всплывающем окне, правильно предположить причину и направить ремонтную бригаду к поврежденному узлу, после чего триггер аварии снимается и ФМ будет функционировать в штатном режиме. Также возможна реализация скрытых аварий, связанных со снижением эффективности работы механизма. Вариант реализации аварий приведен на Рис. 5.

Обсуждение (Discussion). Существует несколько подходов к созданию интерактивных методов обучения персонала, причем у каждого

Рис. 4. Форма панели аварий и блокировок ФМ

Fig. 4. FM alarm and blocking panel form

```

319
320 //Предупреждение и авария по уровню
321 if (level > wrnHilvl) stWrn |= 0x0008;
322 else stWrn &= ~(0x0008);
323
324 if (level > fltHilvl)
325 {
326     if (timeHiLevelStart == 0) timeHiLevelStart = milliseconds;
327     if (milliseconds - timeHiLevelStart > timeHiLevel) stFlt |= 0x0001;
328 }
329 else timeHiLevelStart = 0;
330
331
332 //Авария обратной связи
333 if ((inFltWord & 0x0004) > 0) {
334     if (timeBrokenSensorStart == 0) timeBrokenSensorStart = milliseconds;
335     if (milliseconds - timeBrokenSensorStart > timeBrokenSensor) stFlt |= 0x0004;
336 } else timeBrokenSensorStart = 0;
337
338 //Авария рассогласование пегулятора
339 if ((inFltWord & 0x0040) > 0 && Math.Abs(level - spLevel) > 6 && (state & 0x0002) > 0)
340 {
341     VolumePV -= (float)(random.Next(100)) / 1000;
342     if (timeBadRegStart == 0) timeBadRegStart = milliseconds;
343     if (milliseconds - timeBadRegStart > timeBadReg) stFlt |= 0x0001;
344 }
345 else timeBadRegStart = 0;
346
347
348 //Если есть аварии то в статус записываем бит аварии
349 if ((stFlt & 0x0FFF) > 0) state |= 0x2000;
350 else state &= ~0x2000;
351
352 //Если есть предупреждение то в статус записываем бит предупреждения
353 if (stWrn > 0) state |= 0x1000;
354 else state &= ~0x1000;
355

```

Рис. 5. Пример реализации процедур генерации и определения аварий

Fig. 5. The procedures for generating and determining accidents realization example

из этих подходов есть свои достоинства и недостатки. Например, довольно распространенный «тестовый» подход, состоящий из цикла заданий/вопросов с тестовой системой оценки, направлен в первую очередь на формальную проверку знаний и, зачастую связан с заучиванием заданий без углубления в суть вопросов. Данный подход, пусть и позволяет в краткие сроки создать базовую систему подготовки рабочего персонала, может выступать лишь основой, требуя дальнейшего обучения на практике.

С другой стороны, создание полноценных тренажеров, дублирующих автоматизированное рабочее место оператора (АРМ), требует использования полноценных комплексов промышленного оборудования: покупки промышленных контроллеров, использования лицензированных дистрибутивов совместимых SCADA систем, использования их внутренних инструментов эмуляции, пусть и обладающих широким и продуманным инструментарием, но зачастую не специализированных под конкретную задачу и предназначенных для отладки, но не для эмуляции процесса в целом [17]. Данный подход позволяет создать хороший эмулятор рабочего места, полностью копирующий интерфейс и окружение, но обладающий ограниченным функционалом. Он требует специального оборудования и дорог как при разработке, так и по себестоимости конечного продукта как готовой станции.

В свою очередь, написанный «с нуля» программный продукт, предназначенный для установки на стандартную операционную систему пользовательского ПК, позволяет существенно снизить себестоимость рабочей станции и получить гибкий функционал для реализации всех поставленных задач [18].

Основным преимуществом и в то же время ловушкой данной методологии создания ЭПП, является размытая граница между обобщением и детализацией процесса. Провести черту между достаточным функционалом и избыточной достоверностью зачастую бывает довольно сложно. Действительно, в ходе реализации некоторых классов наша команда сталкивалась с моментами, когда затрачиваемое время на описание определенного эффекта, скажем, вихревого движения пульпы в зумпфе при тангенциальном заходе питающего патрубка, превышало все разумные пределы, но не давало существенного выигрыша в контексте решения общей задачи. С другой стороны, излишнее упрощение ведет к снижению общей достоверности, система перестает отражать реальность протекания технологических процессов и теряет смысл. Более того, это снижает отдачу от процесса обучения персонала, который в кратчайшие сроки приспособливается

к симуляции и процесс обучения на этом завершается, вырождаясь в повторение закриптованных действий. Однако столь гибкий подход при разумном использовании позволяет обходить «узкие места» математических описаний процесса обогащения, позволяя решать задачи существенной сложности в реальные сроки.

Выводы (Findings). Подводя итоги, можно сделать несколько выводов о проделанной работе.

1. Создание полноценного эмулятора SCADA-системы горно-обогатительного предприятия является важной и актуальной задачей, требующей творческого подхода к решению.

2. Дополнительные функции, повышающие общий уровень достоверности и вовлечения обучаемого в процесс, такие как элементы качественного и количественного расчета, интерактивные задания, рандомизированные неисправности и расчет выхода полезного продукта за время тестирования, позволяют добиться более высокого уровня подготовки диспетчерского персонала, а также сократить период обучения и адаптации непосредственно на рабочем месте.

3. Использование системного научного подхода на ранних этапах разработки программного комплекса позволяет существенно сократить время реализации и серьезно упростить математический аппарат моделирования при сохранении общей функциональности кода в рамках поставленной задачи.

СПИСОК ЛИТЕРАТУРЫ

1. Лемента О. Ю. Производственный травматизм как фактор снижения экономической эффективности работы горнодобывающих предприятий: сборник трудов конференции. // Экономика и управление: проблемы, тенденции, перспективы развития: материалы II Междунар. науч.-практ. конф. (Чебоксары, 4 март 2016 г.) / редкол.: О. Н. Широков [и др.] Чебоксары : Центр научного сотрудничества «Интерактив плюс», 2016. С. 115-119.
2. Андриюшкевич С. К., Журавлев С. С., Золотухин Е. П. [и др.] Разработка системы мониторинга с использованием имитационного моделирования // Пробл. информатики. 2010. № 4. С. 65-75.
3. Пьявченко Т. А. Проектирование АСУТП в SCADA-системе. Таганрог : ТТИ ЮФУ, 2007.
4. Ивушкин А. А. Основы создания и внедрения систем автоматизации управления объектами угольной отрасли: Дис. . . . докт. техн. наук. Новокузнецк : ФГБОУ ВПО «СибГИУ», 2007. 312 с.
5. Шаповал И. В., Рузаков В. О., Каримова Л.

К., Мулдашева Н. А., Бейгул Н. А., Ильина Л. А., Волгарева А. Д. Производственный травматизм на горнодобывающих предприятиях республики Башкортостан в 2017-2022 годах // Безопасность и охрана труда. 2024. № 1 (98). С. 8-12.

6. Qiantong Xu, Alexei Baevski, Tatiana Likhomanenko, Paden Tomasello, Alexis Conneau, Ronan Collobert, Gabriel Synnaeve, and Michael Auli. Self-training and pre-training are complementary for speech recognition. In: ICASSP, 2021.

7. Крупенченкова С. А. Производственный травматизм на предприятиях горнодобывающей промышленности // В сборнике: Наука и практика: актуальные вопросы, достижения и инновации. сборник статей VI Международной научно-практической конференции. Пенза, 2025. С. 25-27.

8. Карначев И. П., Челтыбашев А. А., Судак С. Н. Практическая реализация концепции культуры промышленной безопасности на примере крупнейших горных предприятий // Безопасность техногенных и природных систем. 2022. № 3. С. 24-31. DOI: 10.23947/2541-9129-2022-3-24-31. EDN: GYHXYO

9. Mikhailov A. M., Rabeeakh S. K. Trading bots that use machine learning // В сборнике: Наука преобразует реальность. Материалы Международной междисциплинарной научно-практической студенческой конференции. Отв. редактор Е. А. Маклакова. Воронеж, 2023. С. 13-19. DOI: 10.58168/STR2023_13-19. EDN: RHVHZZ

10. Страуструп Бьерн. Язык

программирования C++. Специальное издание. Под ред. Н. Н. Мартынова. М. : БИНОМ, 2001. 1136 с.

11. Солтис М. Введение в анализ алгоритмов СПб. : ДМК. 2019. 269 с.

12. Кормен Т., Лейзерсон Ч. Алгоритмы: построение и анализ, 3-е издание М. : ООО И.Д. Вильямс, 2013. 1328 с.

13. Lange T. Intellegent SCADA System // Engeneer IT. Automation and Technical control. April 2007. P. 26-30.

14. Бергер Г. Автоматизация с помощью программ STEP7 LAD и FBD. Siemens AG, 2001. 605 с.

15. Bishop C. M. Pattern Recognition and Machine Learning // Springer. 2006. 738 p.

16. Оксман В. С., Трубецкой Н. К., Гражданкин А. И. Анализ летальных несчастных случаев в горнорудной и нерудной промышленности России // Безопасность труда в промышленности. 2021. № 3. С. 28.

17. Рыбалев А. Н., Николаец Ф. А. // Разработка и эмулирование АСУ ТП с использованием программ разных производителей и типов // Вестник АмГУ. 2018. 73-82 с.

18. Журавлев С. С. Программно-аппаратный комплекс для тестирования программ управления АСУ ТП шахт и рудников // Вычислительные технологии. Том 18. Специальный выпуск. 2013. 150-155 с.

© 2025 Авторы. Эта статья доступна по лицензии Creative Commons «Attribution» («Атрибуция») 4.0 Всемирная (<https://creativecommons.org/licenses/by/4.0/>)

Авторы заявляют об отсутствии конфликта интересов.

Об авторах:

Липина Галина Александровна, старший преподаватель кафедры высшей математики, Кузбасский государственный технический университет имени Т. Ф. Горбачева, 650000, Россия, г. Кемерово, ул. Весенняя, 28, e-mail: galipina@yandex.ru

Липин Артем Вадимович, кандидат технических наук, главный инженер ООО "Сомерсет Интернэшнл Раша", 143026, Россия, Московская область, г. Одинцово, д. Сколково, ул. Новая, д. 100, e-mail: lipiav@mail.ru.

Заявленный вклад авторов:

Липин Артем Вадимович – сбор и анализ данных, концептуализация исследования, программная реализация.

Липина Галина Александровна – выводы, научный менеджмент, оформление текста, обзор литературы.

Все авторы прочитали и одобрили окончательный вариант рукописи.

Original article

METODOLOGY OF SUPERVISORY CONTROL AND DATA ACQUISITION (SCADA) SYSTEMS EMULATION SOFTWARE DEVELOPMENT

Galina A. Lipina¹,
Artem V. Lipin²

¹ T.F. Gorbachev Kuzbass State Technical University² LCC «Somerset international Russia»

* for correspondence: galipina@yandex.ru

**Article info**

Received:

01 October 2025

Accepted for publication:

15 November 2025

Accepted:

02 December 2025

Published:

22 December 2025

Keywords: emulation, software package, production process, training, mathematical model, mineral processing, coal industry, flotation.

Abstract.

In this article discusses the relevance and methodology of development a software package for emulating supervisory control and data acquisition (SCADA) systems for training dispatch personnel. This software package is part of the digitalization of production, aiming to improve process efficiency and reduce associated costs by reducing the number of emergencies and overall downtime. Using the example of development one of the software package's elements (classes)—namely, a flotation machine emulator, a common type of mining and processing equipment—the article describes the step-by-step process for implementing this structural unit, and then the overall concept of the system's operation. The approach to class creation is described in several stages, such as: defining the key operational and system variables, defining the laws governing their interaction, defining generalizing the physical processes occurring during the operation of process equipment to obtain a reliable picture of its operation using a simplified mathematical framework. Although this article discusses only a general approach to developing a software system, without reference to a specific programming language, it draws on the implementation process of an existing software product and summarizes the experience gained in creating a workable model. The article includes screenshots of the software product. The realization of this idea can vary significantly, as can the level of abstraction when creating classes, and depends primarily on the goals and objectives of the developer.

For citation: Lipina G.A., Lipin A.V. Metodology of supervisory control and data acquisition (SCADA) systems emulation software development. *Vestnik Kuzbasskogo gosudarstvennogo tekhnicheskogo universiteta*=Bulletin of the Kuzbass State Technical University. 2025; 6(172):104-113. (In Russ., abstract in Eng.). DOI: 10.26730/1999-4125-2025-6-104-113, EDN: KFKZAK

REFERENCES

1. Lementa O.Yu. Proizvodstvennyj travmatizm kak faktor snizheniya ehkonomicheskoy ehffektivnosti raboty gornodobyvayushhih predpriyatij: sbornik trudov konferentsii. // Ehkonomika i upravlenie: problemy, tendentsii, perspektivy razvitiya : materialy II Mezhdunar. nauch.-prakt. konf. (Cheboksary, 4 mart 2016 g.) / redkol.: O.N. Shirokov [i dr.]. Cheboksary: Tsentr nauchnogo sotrudnichestva «Interaktiv plyus»; 2016. S. 115-119.
2. Andryushkevich S.K., Zhuravlyov S.S., Zolotukhin E.P. i dr. Razrabotka sistemy monitoringa s ispol'zovaniem imitatsionnogo modelirovaniya. *Probl. informatiki*. 2010; 4:65-75.
3. P'yavchenko T.A. Proektirovanie ASUTP v SCADA-sisteme. Taganrog: TTI YUFU; 2007.
4. Ivushkin A.A. Osnovy sozdaniya i vnedreniya sistem avtomatizatsii upravleniya ob'ektami ugol'noj otrasli: Dis. . . . dokt. tekhn. nauk. Novokuznetsk; FGBOU VPO «SibGIU»; 2007. 312 s.
5. Shapoval I.V., Ruzakov V.O., Karimova L.K., Muldasheva N.A., Bejgul N.A., Il'ina L.A., Volgareva A.D. Proizvodstvennyj travmatizm na gornodobyvayushhih predpriyatiyakh respubliki Bashkortostan v 2017-2022 godakh. *Bezopasnost' i okhrana truda*. 2024; 1(98):8-12.
6. Qiantong Xu, Alexei Baevski, Tatiana Likhomanenko, Paden Tomasello, Alexis Conneau, Ronan Collobert, Gabriel Synnaeve, and Michael Auli. Self-training and pre-training are complementary for speech recognition. *In: ICASSP*. 2021.
7. Krupenchenkova S.A. Proizvodstvennyj travmatizm na predpriyatiyakh gornodobyvayushhej promyshlennosti. *V sbornike: Nauka i praktika: aktual'nye voprosy, dostizheniya i innovatsii. sbornik statej VI Mezhdunarodnoj nauchno-prakticheskoy konferentsii*. Penza, 2025. S. 25-27.
8. Karnachev I.P., Cheltybashev A.A., Sudak S.N. Prakticheskaya realizatsiya kontseptsii kul'tury promyshlennoj bezopasnosti na primere krupnejshikh gornykh predpriyatij. *Bezopasnost' tekhnogennykh i prirodnnykh sistem*. 2022; 3:24-31. DOI: 10.23947/2541-9129-2022-3-24-31. EDN: GYHXYO
9. Mikhailov A.M., Rabeeakh S.K. Trading bots that use machine learning. *V sbornike: Nauka preobrazhaet real'nost'. Materialy Mezhdunarodnoj mezhdistsiplinarnoj nauchno-prakticheskoy studencheskoj konferentsii*. Otv. redaktor E.A. Maklakova. Voronezh, 2023. S. 13-19. DOI: 10.58168/STR2023_13-19. EDN: RHVHZZ
10. Strastrup B'ern. Yazyk programmirovaniya C++. Spetsial'noe izdanie. Pod red. N.N. Martynova. M.: BINOM; 2001. 1136s.

11. Soltis M. Vvedenie v analiz algoritmov SPb.: DMK; 2019. 269 s.
12. Kormen T., Lejzerson Ch. Algoritmy: postroenie i analiz, 3-e izdanie. M.: OOO I.D. Vil'yams; 2013. 1328 s.
13. Lange T. Intelligent SCADA System. *Engineer IT. Automation and Technical control*. April 2007. Pp. 26-30.
14. Berger G. Avtomatizatsiya s pomoshh'yu programm STEP7 LAD i FBD. *Siemens AG*. 2001. 605 s.
15. Bishop C.M. Pattern Recognition and Machine Learning. *Springer*. 2006. 738 s.
16. Oksman V.S., Trubetskoj N.K., Grazhdankin A.I. Analiz letal'nykh neschastnykh sluchaev v gornorudnoj i nerudnoj promyshlennosti Rossii. *Bezopasnost' truda v promyshlennosti*. 2021; 3:28.
17. Rybalev A.N., Nikolaets F.A. Razrabotka i ehmulirovanie ASU TP s ispol'zovaniem programm raznykh proizvoditelej i tipov. *Vestnik AmGU*. 2018. 73-82s.
18. Zhuravlyov C.C. Programmno-apparatnyj kompleks dlya testirovaniyaprogramm upravleniya ASU TP shakht i rudnikov. *Vychislitel'nye tekhnologii*. T. 18. Spetsial'nyj vypusk. 2013. Pp. 150-155.

© 2025 The Authors. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

The authors declare no conflict of interest.

About the authors:

Galina A. Lipina, senior lecturer of the Department of Mathematics, T.F. Gorbachev Kuzbass State Technical University, 28 street Vesennyaya, Kemerovo, 650000, Russian Federation, 7676-5937, e-mail: galipina@yandex.ru

Artem V. Lipin, chief engineer of LCC «Somerset international Russia» 143026, Moscow region, Odintsovo, Skolkovo, Novaya st., bldg. 100, e-mail: lipiav@mail.ru.

Contribution of the authors:

Artem V. Lipin – data collection and analysis, conceptualization of research, software implementation.

Galina A. Lipina – conclusions, scientific management, text design, literature review..

All authors have read and approved the final manuscript.

