



УДК 004.94
doi:10.21685/2587-7704-2022-7-2-3



Open
Access

RESEARCH
ARTICLE

Проектирование системы SCADA: предложение по оптимизации производственной линии

Анастасия Александровна Казаченко

Пензенский государственный университет, Россия, г. Пенза, ул. Красная, 40
ms.kazachenko@mail.ru

Евгения Анатольевна Данилова

Пензенский государственный университет, Россия, г. Пенза, ул. Красная, 40
siori@list.ru

Аннотация. Освещен вопрос оптимизации производственной линии путем внедрения системы SCADA. Рассмотрены основные компоненты системы и механизмы взаимодействия между ними. Определена архитектура модели CIM, в которой выявлены уровни производственного процесса и его иерархия, а также описаны элементы оборудования и программного обеспечения для определения входных данных и результатов обработки данных.

Ключевые слова: хранилище данных, SCADA, оптимизация производства, технический интерфейс, производственный процесс, управление производством

Для цитирования: Казаченко А. А., Данилова Е. А. Проектирование системы SCADA: предложение по оптимизации производственной линии // Инжиниринг и технологии. 2022. Т. 7(2). С. 1–9. doi:10.21685/2587-7704-2022-7-2-3

SCADA system design: production line optimization proposal

Anastasiya A. Kazachenko

Penza State University, 40 Krasnaya Street, Penza, Russia
ms.kazachenko@mail.ru

Evgeniya A. Danilova

Penza State University, 40 Krasnaya Street, Penza, Russia
siori@list.ru

Abstract. The article highlights the issue of optimizing the production line by implementing the SCADA system. The main components of the system and mechanisms of interaction between thereof are considered. The architecture of the CIM model is defined, the levels of the production process and its hierarchy are specified, and the hardware and software elements for determining the input data and data processing results are described.

Keywords: data storage, SCADA, production optimization, technical interface, production process, production management

For citation: Kazachenko A.A., Danilova E.A. SCADA system design: production line optimization proposal. *Inzhiniring i tekhnologii = Engineering and Technology*. 2022;7(2):1–9. (In Russ.). doi:10.21685/2587-7704-2022-7-2-3

Введение

Линейное производство превратилось в логистику производственных процессов. Теоретическая идея сфокусирована на тейлоризме [1], который применен на практике Рэнсоном Олдсом, открывшим самую первую производственную линию для создания мануфактурного автомобильного производства Curve Dash [2]. С тех пор промышленность изменила способ производства продукции, задачи были разделены для получения лучших результатов, а также были внедрены достижения в web-технологиях, аппаратном и программном обеспечении, которые позволяют достичь максимальной производительности и качества на производственных линиях.

В отрасли конкурентное преимущество обычно получают компании, которые используют технологии для автоматизации своих производственных линий [3], поэтому это стало тенденцией. Учи-



тывая сложность производства, а также использование огромного количества данных на разных этапах, организации вынуждены объединять информацию на всех уровнях, которые в то же время включают в себя различные системы, такие как планирование ресурсов предприятия, система управления производством, диспетчерский контроль и сбор данных [4]. Благодаря достижениям в Интернете – доступности данных из любого места и с любого устройства – появилась возможность применения концепции Интернета-вещей для промышленных процессов [5]. Имеется в виду интеллектуальная индустрия, четвертая промышленная революция, которую обычно называют Индустрия 4.0 [4, 6, 7]. Программное обеспечение для автоматизации производства часто бывает дорогостоящим и трудным для реализации [8]. Это представляет проблему для начинающих компаний, которые сталкиваются с потребностью в нем, что приводит к сокращению их производства, поскольку оно не считалось необходимым с самого начала [9].

Архитектура проекта графически представляет физические компоненты систем и их взаимосвязь. Системы контроля и надзора созданы на основе модели, предложенной французским инженером Жаном Батистом Вальдером под названием Computer Integrated Manufacturing (CIM), которая необходима для определения иерархической и функциональной структуры системы [10–12]. Рисунок 1 иллюстрирует типичную пирамиду CIM, которая выделяет основные 4 уровня, присутствующие в большинстве производственных систем. Полевой уровень 0 включает устройства (датчики и исполнительные механизмы – приводы), которые собирают данные о физических переменных и отправляют их на уровень обработки 1. На уровне обработки 1 есть устройства, такие как PLC, RTU, PAC и PC, анализирующие и обрабатывающие данные для отправки координатору (уровень ячейки 2), который отвечает за наблюдение и контроль над процессами (система SCADA), генерируя соответствующую информацию для отслеживания и управления процессом на верхних уровнях. Уровень цеха 3 отвечает за планирование и производство продукции (системы MRP и MES), в то время как последний уровень 4 берет информацию из процессов, которая должна быть проанализирована в информационных системах компании (ERP) [11, 13].

Различные системы и / или устройства, которые используются при автоматизации производственной линии на разных уровнях, представленных выше:

L4: ERP (планирование ресурсов предприятия);

L3: MES (система производственного предприятия) и MRP (планирование потребности в материалах);

L2: SCADA (Supervisory Control and Data Acquisition);

L1: Программируемые логические контроллеры (ПЛК), DCS;

L0: Датчики и исполнительные механизмы.

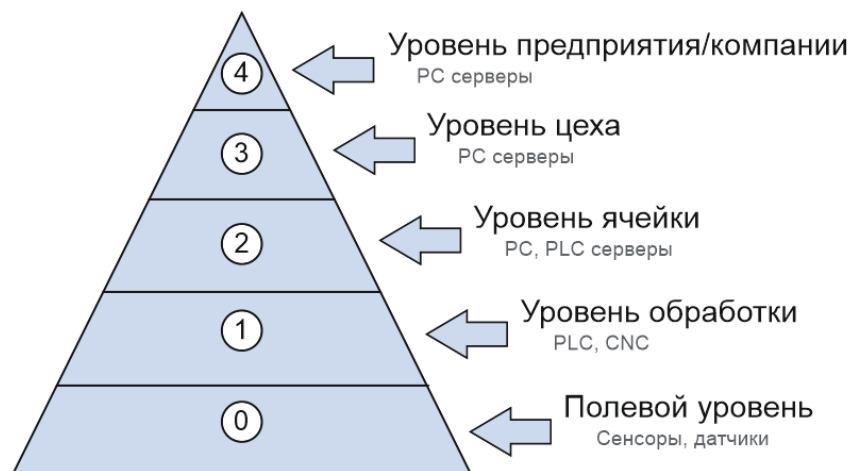


Рис. 1. CIM пирамида

Модель CIM может варьироваться в зависимости от потребностей. Анализ на уровнях компьютерной модели, интегрированной для производства, дает общую картину для создания архитектуры предлагаемой системы.

Существуют различные решения по оптимизации производственных процессов, наиболее распространенными являются SCADA и DCS (распределенная система управления). Первая использует



ся для сбора данных, а вторая фокусируется на процессе. Существуют также различия между затратами и типом отрасли, в которой они работают.

В данной статье рассмотрен пример реализации системы SCADA с использованием технологий Freeware (бесплатный софт) и Open Source (технологии с открытым исходным кодом) [8, 14]. Еще одно преимущество системы – это возможность взаимодействия между системами на других уровнях производства. Представленная архитектура сокращена, но все же эффективна, так как она может обеспечить решение основных потребностей автоматизированной производственной линии.

Работа организована следующим образом: в разделе 1 определена архитектура модели CIM. В разделе 2 рассмотрены механизмы связи между различными уровнями с использованием облегченных протоколов для достижения высокой производительности и доступности. В разделе 3 представлен уровень данных с использованием хранилищ данных. Наконец, сделаны выводы.

1. Архитектура и дизайн системы

А. Архитектура системы.

Ранее говорилось об уровнях внутри производственного процесса, на которых можно наблюдать длительный процесс систем, чтобы перейти к планированию ресурсов компании. Далее представлена архитектура для системы SCADA:

- 1) телеметрическая система, отвечающая за область сбора устройств;
- 2) система SCADA;
- 3) уровень взаимосвязи данных на верхнем уровне.

На рис. 2 представлена схема предлагаемой архитектуры. Наблюдается сокращение архитектуры по сравнению с традиционной.

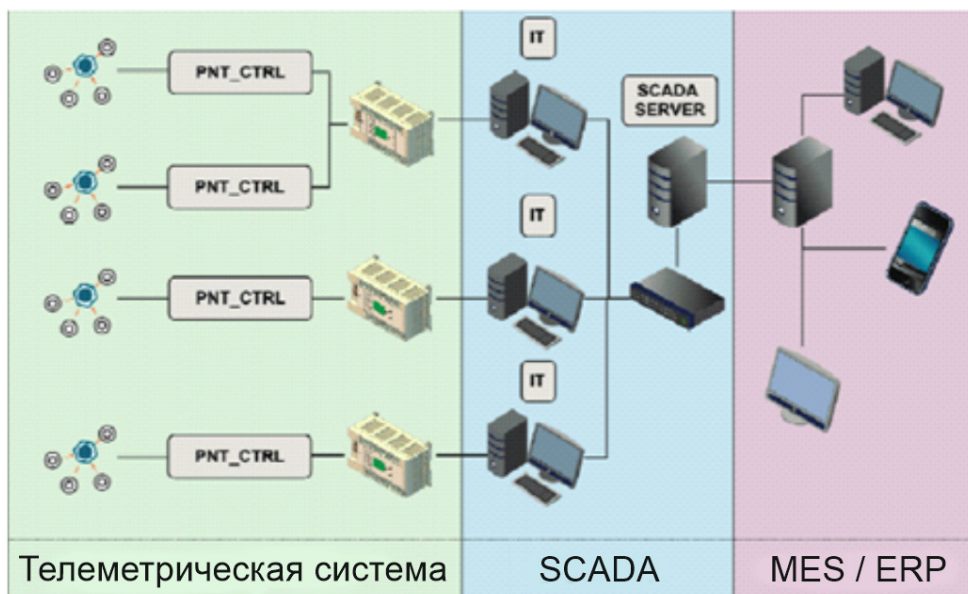


Рис. 2. Предложенная архитектура системы

Б. Дизайн системы.

В этом разделе будут определены требования, которым должна удовлетворять система, в основном выявляя программные и аппаратные средства, которые будут участвовать в реализации. Следуя принципам Тейлора [15], рассмотрено разделение задач, которые разграничиваются как этапы управления (PTS_CTRL). В основном PTS_CTRL представляют собой подпроцесс производственной линии, который имеет свои особенности, с помощью которых мы обращаемся к различным устройствам процесса, различным входам и выходам данных. Этапы управления не зависят от телеметрической системы, поскольку являются виртуальными, они реализованы в виде программного обеспечения на небольших серверах, которые мы определяем как технический интерфейс (ТИ).

Иерархически каждый ТИ является главой устройств процесса (PLC, PAC, Arduino, среди прочих), которые будут отправлять информацию, закодированную с определенной структурой, которая определяет, к какому PTN_CTRL принадлежит. Здесь происходит виртуализация – ТИ берет информацию и добавляет дополнительные данные, включая идентификацию, которая поможет координато-



ру в интеграции компонентов. Этот координатор реализуется через программное обеспечение и является подчиненным сервером SCADA, а тот, в свою очередь, является подчиненным устройством следующих систем на верхних уровнях, таких как MES, MRP и / или ERP, интегрирующих информацию для всех уровней (см. рис. 2).

Обмен данными внутри предлагаемой системы и телеметрической системы состоит из считывания данных с использованием интерфейса, например, последовательного порта. Технический интерфейс TI возьмет на себя эту задачу. Сервер SCADA связывается с системами верхнего уровня через веб-серверы или хранилище данных. В то же время информация собирается на верхних уровнях для добавления метаданных к мониторингу и отслеживанию продукта на производственной линии. Тот вариант реализации, который предлагается в данной статье, объединяется тремя элементами: службами связи, службами хранения данных между системами и пользовательским интерфейсом.

В. Пользовательский интерфейс.

Пользовательский интерфейс разрабатывается путем качественного проектирования графического интерфейса пользователя (GUI), который реализован с помощью набора веб-технологий. Большая часть бизнес-логики реализована с помощью PHP-сервера и JavaScript (на стороне клиента), в дополнение к языку CSS для улучшения представления [21]. Графический интерфейс пользователя (GUI) можно разделить на три основные составляющие: модуль визуализации, модуль настройки и модуль управления (CRUD).

На рис. 3 показан набор модулей, составляющих графический интерфейс, модули CRUD (создание, чтение, обновление и удаление) практически выполняют задачу управления системными каталогами, чтобы снабжать базу данных физическими и виртуальными компонентами, составляющими производственную линию. Это помогает в работе сервера SCADA, который в основном включает логистику для руководства процессом, координирует подчиненные узлы (ИТ) и хранит соответствующие данные. Организационные действия SCADA-сервера создают большую нагрузку в этой области (обработка на уровне бизнес-логики), удаляя трафик данных в критических областях (в ИТ и контрольных точках).

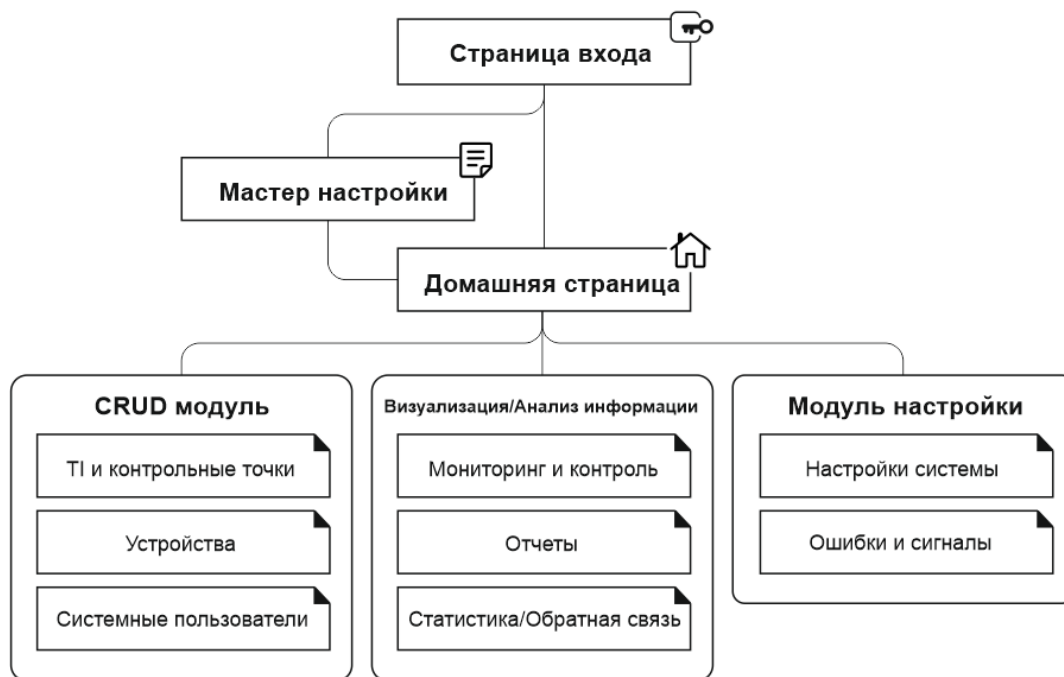


Рис. 3. Структура SCADA-системы

Модули настройки можно использовать для перенастройки системы, будь то масштабируемость или просто необходимость замены или обновления компонентов. Еще одна важная особенность – это настройка отображения ошибок и последующих за ними аварийных ситуаций. Модули визуализации и статистики представляют собой основную часть системы, в которой осуществляется мониторинг и управление предприятием, а также ведется статистика и использование информации посредством отчетов.



2. Связь между системами

Связь между различными уровнями производственной линии должна быть быстрой и надежной, чтобы выдержать нагрузку в реальном времени, поэтому используются технологии, отвечающие требованиям реализации. Ниже определены механизмы связи между элементами, которые взаимодействуют с системой SCADA; а также дано краткое введение в работу с уровнем данных.

На рис. 4 показана схема связи. Связь делится на три категории, представленные сокращенными уровнями архитектуры:

- 1) связь между системой телеметрии и ТИ;
- 2) второй уровень связывает технический интерфейс с сервером SCADA;
- 3) связь SCADA с модулем мониторинга и контроля.

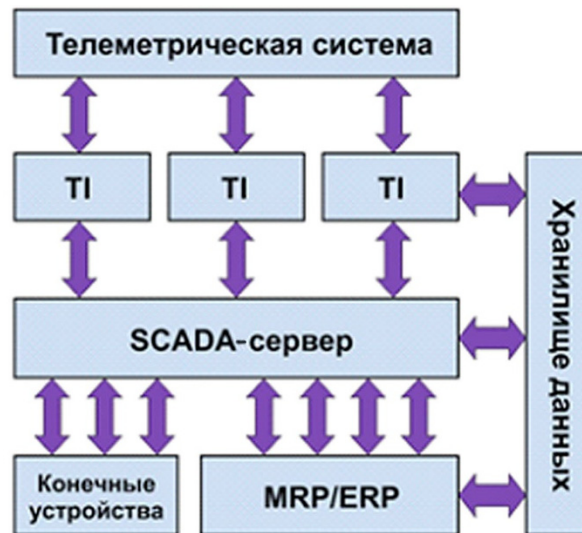


Рис. 4. Связь между уровнями производственных процессов

Сервер SCADA собирает информацию со всех технических интерфейсов для обновления хранилища базы данных с помощью систем более высокого уровня. Этот механизм будет подробно описан в следующем разделе.

Сервисы для конечных устройств оснащены доступом к информации о состоянии производственной линии без запроса напрямую от систем управления на более высоких уровнях. Таким образом, реализуются все необходимые меры безопасности.

А. Связь между системой телеметрии и техническим интерфейсом.

Основная задача телеметрической системы – дистанционное измерение физических величин, которые затем отправляются в центр управления. Центр управления – это технический интерфейс, который реализован на основе синтаксиса, отвечающего требованиям приема и отправки данных на этом этапе. Связь с системой телеметрии обеспечивается получением закодированных строк посредством интерфейса связи цифровых данных. Эти строки содержат данные от полевых устройств и информацию о состоянии производственного процесса, к тому же они могут включать сообщения оповещения, предупреждения и сигналы об ошибках. Строка кода, которая будет обрабатываться интерфейсом ТИ в каждом описанном выше случае, приведена ниже.

1. Получение данных.

Получение данных основано на закодированной строке; данные каждого устройства разделяются знаком «,» (если этот знак используется в записи, для разделения может быть использован другой символ, например, «/ n» новой строки), значения остаются в скобках «[]», и каждое из них отделяется знаком «|», как представлено ниже:

[type_msg | id_dev | value_dev], [type_msg | id_dev | value_dev].

Компоненты, составляющие строку:

- type_msg: всегда должно быть «dt», указывающее, что ТИ является устройством информационной телеметрической системы;
- id_dev: это поле соответствует идентификатору устройства, который присваивается системой SCADA в модуле конфигурации, с этим идентификатором сервер ТИ и SCADA знают, к какой группе или контрольной точке раздела принадлежит устройство;



- *value_dev*: единица измерения; максимальное, минимальное и оптимальное значения указываются в модуле конфигурации системы SCADA.

2. Получение сообщений о событиях.

Сообщения о событиях помогают запускать предупреждения или отправлять сообщения с информацией с полевых устройств. Эта информация помогает при принятии решений в производственном процессе. Независимо от того, где реализован контроль физических величин устройств, уведомления, ошибки и предупреждения могут быть отправлены. Кодировка строки следующая:

[*type_msg* | *id_dev* | *code_evt*], [*type_msg* | *id_dev* | *code_evt*]

- *type_msg*: типы событий включают: «*er*» для запуска событий ошибки, «*wr*» для запуска предупреждений и «*nt*» для уведомлений;

- *code_evt*: это число, представляющее код события, определенный в каталоге системы SCADA, эти коды используются для отображения деталей происходящих событий.

3. Отправление данных телеметрической системе.

Связь с телеметрической системой очень важна, потому что в любой момент ТИ может отправлять сигналы для активации исполнительных механизмов. Реализация аналогична получению данных; в этом случае информационный пакет состоит из типа действия (*type_act*), идентификатора устройства, выполняемого действия (*exe_act*) и списка параметров (*param_act*). Кодировка строки следующая:

[*type_act*|*id_dev*|*exe_act*|*param_act*], [*type_act*|*id_dev*|*exe_act*|*param_act*]

Б. Связь между техническим интерфейсом и SCADA-сервером.

На этом уровне связи группа данных добавляется в качестве идентификатора ТИ и данных от каждого устройства: группа идентификаторов, раздела и контрольной точки. На этот раз кодировка представляет собой объект типа JSON (JavaScript Object Notation), облегченный формат для обмена данными [16]. Закодированный объект JSON отправляется на сервер SCADA по протоколу WebSocket. На этом уровне ТИ становится клиентом, поэтому реализовывать протокол связи WebSocket должно программное обеспечение. На рис. 5 показана эта схема связи. Можно увидеть формы, которые принимают объекты JSON для отправки на сервер SCADA. Каждый объект содержит набор контрольных точек, которые могут иметь одну или несколько секций. В каждой секции может быть одна или несколько групп устройств, и наконец, добавляется информация о каждом устройстве.

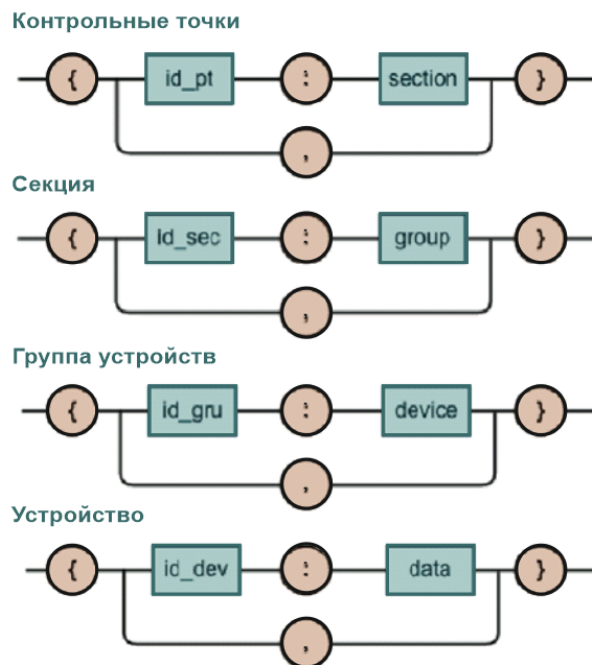


Рис. 5. JSON, закодированный для отправки на SCADA-сервер

С учетом вышеуказанных требований создается фоновый процесс – демон (UNIX) / служба (Windows) для отправки данных. Учитывая, что WebSockets работает через TCP, создается гнездо TCP, которое адаптируется для связи с сервером.



В. Связь SCADA-сервера с модулем мониторинга и контроля.

Связь между сервером и модулем мониторинга и управления SCADA осуществляется с помощью протокола WebSocket. Сервер SCADA настроен как сервер WebSockets, модуль мониторинга и управления выступает в качестве клиента, являясь веб-приложением для браузера. В настоящее время протокол WebSocket широко поддерживается в браузерах, включая старые версии Internet Explorer, благодаря библиотеке SocketIO, поэтому проблем с совместимостью нет [197]. Базовый шаблон использования – создать экземпляр объекта и снабдить код алгоритмами для события OnOpen при запуске соединения, OnMessage – когда информация получена, OnError – для обработки ошибок, и OnClose – для реализации действий по закрытию соединения. Объекты JSON, полученные веб-клиентом, декодируются для обновления графического интерфейса пользователя [18].

3. Слой данных с использованием хранилища данных

Сохранение данных в системе SCADA имеет большое значение для обратной связи процессов, отчетности, интеллектуального анализа данных и других функций.

Хранилище данных – это корпоративная база данных, которая характеризуется интеграцией и отладкой информации из одного или нескольких источников.

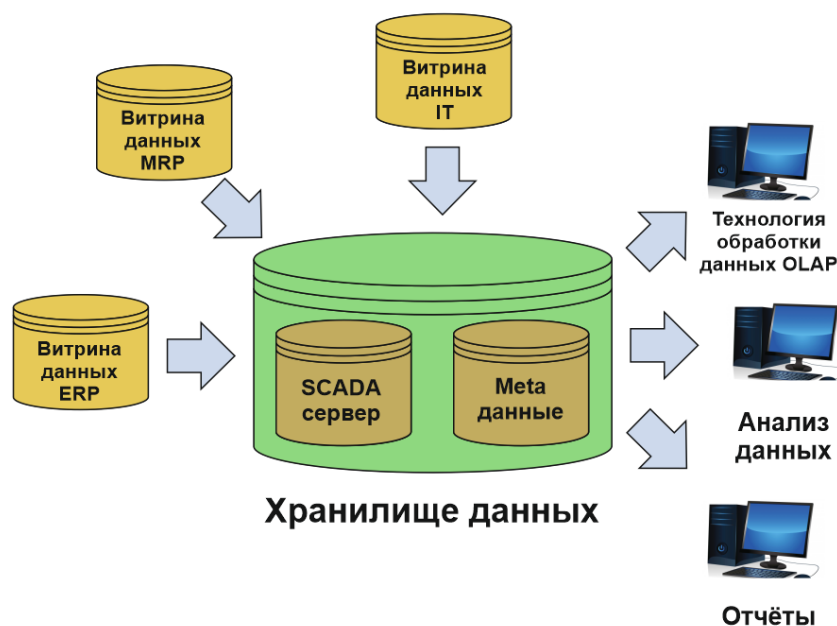


Рис. 6. Предложенное хранилище данных

Каждая система, участвующая в процессах промышленного производства, имеет базу данных, с точки зрения хранилища данных, которую они называют рынком данных [20]. Так извлекается сводная информация, а затем она обрабатывается различными механизмами, такими как интеллектуальный анализ данных, OLAP (Online Analytical Processing) или создание отчета. На рис. 6 показано предлагаемое хранилище данных. Внедрение хранилища данных предполагает посредника между системами (MES, MRP, ERP и т.д.), которые взаимодействуют с производственной линией. Это решение ограничено процессами системы SCADA, однако может оставаться открытым для других приложений. База данных, используемая системой SCADA, является частью описанного выше хранилища данных. На рис. 7 показан простой вид модели данных системы SCADA.

Основной раздел базы данных системы ориентирован на создание технических интерфейсов с соответствующими контрольными точками, которые разделены на секции. Другой важный раздел – определить, какие группы устройств связаны с секциями, например, тостер с четырьмя датчиками температуры имеет одну или несколько групп исполнительных механизмов для управления ими. Разделение ТП на контрольные точки помогает определить, какая информация отображается для наблюдающего пользователя.

Заключение

Технологии все чаще используются в различных сферах деятельности, промышленность не является исключением. Для повышения конкурентоспособности и изготовления качественной продук-



ции в настоящее время предприятия вынуждены оптимизировать производственные процессы, и данное решение – внедрение SCADA-систем – позволяет достичь наилучших результатов.

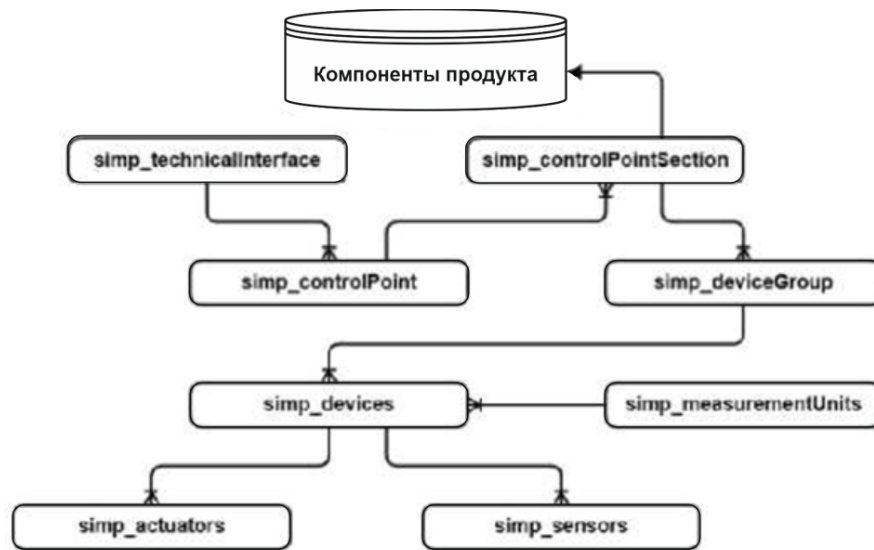


Рис. 7. Модель базы данных SCADA-системы

SCADA-система служит для уменьшения иерархической схемы производственных процессов, повышения операционного и функционального уровня, а также для обеспечения схемы связи, которая улучшит производительность и облегчит взаимодействие между системами. А внедрение экспертных систем, таких как многоагентные системы (MAS), может предоставить производственной линии больше возможностей для процесса обратной связи с использованием базы знаний.

Список литературы

1. Taylorism. Scientific management system. Encyclopedia Britannica. URL: <http://global.britannica.com/topic/Taylorism> (дата обращения: 22.09.2015).
2. Dutton K. A brief history of the car // New Ideas. 2006. Vol. 1.
3. Best M. H. The New Competition: Institutions of Industrial Restructuring. Cambridge (Mass.) : Harvard University Press, 1990. 296 p.
4. Polonia P. V., Bier Melgarejo L. F. and Queiroz M. H. D. A resource oriented architecture for Web-integrated SCADA applications // IEEE World Conference on Factory Communication Systems (WFCS). 2015. P. 1–8.
5. Waher P. Learning Internet of Things. Packt Publishing Ltd, 2015. 242 с.
6. Adeyeri M. K., Mporfu K. and Adenuga O. T. Integration of agent technology into manufacturing enterprise: A review and platform for industry 4.0 // 2015 International Conference on Industrial Engineering and Operations Management (IEOM). 2015. P. 1–10.
7. Gorecky D., Schmitt M., Loskyll M., Zuhlke D. Human-machine-interaction in the industry 4.0 era // 2014 12th IEEE International Conference on Industrial Informatics (INDIN). 2014. P. 289–294.
8. Almas M. S., Vanfretti L., Lovlund S., Gjerde J. O. Open source SCADA implementation and PMU integration for power system monitoring and control applications // 2014 IEEE Power & Energy Society General Meeting. 2014. P. 1–5.
9. Fayad M. E., Laitinen M., Ward R. P. Thinking Objectively: Software Engineering in the Small // Commun ACM. 2000. Vol. 43, № 3. P. 115–118.
10. Silverstre-Blanes J., Sempere-Paya V.-M., Albero-Albero T. VAN Applied to Control of Utilities Networks – Requirements and Capabilities // Factory Automation / by J. Silvestre-Blanes. InTech, 2010.
11. Penin A. R. Sistemas SCADA – QuiaPractica. Marcombo, 2007.
12. Zuge C. T. Y., Pereira S. L., Dias E. M. Integration of Information Technology and Automation: Facilitators and Barrier // Wseas Transactions on Systems and Control. 2010. Vol. 5 (5). P. 372–381.
13. Aleaga Loaiza L. F. Diseño de un Sistema de Telemetria y Telecontrol (SCADA) para la Red de Distribution de Agua Potable de la Ciudad de Loja. Ecuador, 2008.
14. Davis R. PLC vs DCS in process control-the distinction is fading // Proceedings IEEE 1992 Annual Textile, Fiber and Film Industry Technical Conference. 1992. P. 6/1–6/5.
15. Jurgens U., Malsch T., Dohse K., Breaking from Taylorism: Changing Forms of Work in the Automobile Industry. Cambridge : Cambridge University Press, 1993.
16. JSON.org. 2013. URL: <http://www.json.org> (дата обращения: 01.09.2015).
17. SOCKET.IO IS OPEN-SOURCE (MIT), Socket.IO. URL: <http://socket.io/> (дата обращения: 01.09.2015).



18. Fette I., Melnikov A. The WebSocket Protocol. URL: <https://tools.ietf.org/html/rfc6455> (дата обращения: 02.10.2015).
19. Sinnexus, Datawarehouse, Sinnexus Business Intelligence Informatica estrategica. URL: http://www.sinnexus.com/business_intelligence/datawarehouse.aspx (дата обращения: 07.07.2015).
20. Reeves L. A Manager's Guide to Data Warehousing. John Wiley & Sons, 2009.
21. Godbolt M. Front-End Architecture: A Modern Blueprint for Scalable and Sustainable Design Systems. O'Reilly Media, 2016.

References

1. Taylorism. Scientific management system. *Encyclopedia Britannica*. Available at: <http://global.britannica.com/topic/Taylorism> (accessed 22.09.2015).
2. Dutton K. A brief history of the car. *New Ideas*. 2006;1.
3. Best M.H. *The New Competition: Institutions of Industrial Restructuring*. Cambridge (Mass.): Harvard University Press, 1990:296.
4. Polonia P.V., Bier Melgarejo L.F. and Queiroz M.H.D. A resource oriented architecture for Web-integrated SCADA applications. *IEEE World Conference on Factory Communication Systems (WFCS)*. 2015:1–8.
5. Waher P. *Learning Internet of Things*. Packt Publishing Ltd, 2015:242.
6. Adeyeri M.K., Mporfu K. and Adenuga O.T. Integration of agent technology into manufacturing enterprise: A review and platform for industry 4.0. *2015 International Conference on Industrial Engineering and Operations Management (IEOM)*. 2015:1–10.
7. Gorecky D., Schmitt M., Loskyll M., Zuhlke D. Human-machine-interaction in the industry 4.0 era. *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*. 2014:289–294.
8. Almas M.S., Vanfretti L., Lovlund S., Gjerde J.O. Open source SCADA implementation and PMU integration for power system monitoring and control applications. *2014 IEEE Power & Energy Society General Meeting*. 2014:1–5.
9. Fayad M.E., Laitinen M., Ward R.P. Thinking Objectively: Software Engineering in the Small. *Commun ACM*. 2000;43(3):115–118.
10. Silverstre-Blanes J., Sempere-Paya V.-M., Albero-Albero T. VAN Applied to Control of Utilities Networks – Requirements and Capabilities. *Factory Automation*, 2010.
11. Penin A.R. *Sistemas SCADA – QuiaPractica*. Marcombo, 2007.
12. Zuge C.T.Y., Pereira S.L., Dias E.M. Integration of Information Technology and Automation: Facilitators and Barrier. *Wseas Transactions on Systems and Control*. 2010;5(5):372–381.
13. Aleaga Loaiza L.F. *Diseno de un Sistema de Telemetria y Telecontrol (SCADA) para la Red de Distribution de Agua Potable de la Ciudad de Loja*. Ecuador, 2008.
14. Davis R. PLC vs DCS in process control-the distinction is fading. *Proceedings IEEE 1992 Annual Textile, Fiber and Film Industry Technical Conference*. 1992:6/1–6/5.
15. Jurgens U., Malsch T., Dohse K., *Breaking from Taylorism: Changing Forms of Work in the Automobile Industry*. Cambridge: Cambridge University Press, 1993.
16. *JSON.org*. 2013. Available at: <http://www.json.org> (accessed 01.09.2015).
17. *SOCKET.IO IS OPEN-SOURCE (MIT), Socket.IO*. Available at: <http://socket.io/> (accessed 01.09.2015).
18. Fette I., Melnikov A. *The WebSocket Protocol*. Available at: <https://tools.ietf.org/html/rfc6455> (accessed 02.10.2015).
19. Sinnexus, Datawarehouse, Sinnexus Business Intelligence Informatica estrategica. Available at: http://www.sinnexus.com/business_intelligence/datawarehouse.aspx (accessed 07.07.2015).
20. Reeves L. *A Manager's Guide to Data Warehousing*. John Wiley & Sons, 2009.
21. Godbolt M. *Front-End Architecture: A Modern Blueprint for Scalable and Sustainable Design Systems*. O'Reilly Media, 2016.

Поступила в редакцию / Received 11.03.2022

Поступила после рецензирования и доработки / Revised 17.04.2022

Принята к публикации / Accepted 05.05.2022