

3. Левчук И.Л. Идентификация математической модели процесса каталитического риформинга на базе нейросетевых технологий / И.Л. Левчук // Математичне моделювання. – Дніпродзержинськ. – 2012. – №2. – С. 77-80.
4. Левчук И.Л. Разработка математической модели процесса каталитического риформинга в каскаде реакторов / И.Л. Левчук // Збірник наукових праць НГУ. – 2012. – №39. – С. 122 – 127.

*Рекомендовано к публикации д.т.н. Слесаревим В.В.
Поступила в редакцию 16.01.15*

УДК 629.78.018

©И.Л. Левчук, Е.В. Белоброва, В.И. Корсун

ПРИНЦИПЫ ИНТЕГРАЦИИ СПЕЦИАЛЬНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННО УПРАВЛЯЮЩИХ СИСТЕМ В СОВРЕМЕННЫЕ SCADA СИСТЕМЫ

В работе исследована проблема интеграции специального программного обеспечения, реализованного на языках программирования высокого уровня, в современные SCADA системы. Разработана обобщенная структура взаимодействия и произведена экспериментальная апробация предложенного способа интеграции на примере SCADA TRACE-MODE 6.

У роботі досліджена проблема інтеграції спеціального програмного забезпечення, реалізованого на мовах програмування високого рівня, в сучасні SCADA системи. Розроблено узагальнену структуру взаємодії спеціального програмного забезпечення з SCADA. Проведена експериментальна апробация запропонованого способу інтеграції на прикладі SCADA TRACE-MODE 6.

We have studied the problem of the integration of special software implemented on high level programming languages, in modern SCADA system. The generalized structure of the interaction of special software with SCADA and experimental validation of the proposed method on the example of the integration of SCADA TRACE-MODE 6.

Вступление. Современные АСУТП представляют собой сложные информационно-управляющие системы, функционал которых реализуется с использованием средств микропроцессорной вычислительной техники и соответствующего общего программного обеспечения. Решение задач верхнего уровня АСУТП, таких как оптимальное управление технологическими процессами по математическим моделям, требует разработки специального программного обеспечения которое, как правило, реализуется на распространенных языках программирования высокого уровня (Delphi, C++, VB). На определенном этапе перед разработчиками встает проблема интеграции общего и специального программного обеспечения в рамках единого комплекса, обеспечивающего функционирование системы управления.

Анализ исследований и публикаций. Область применения АСУТП значительно расширилась в последние годы, благодаря прогрессу в области вычислительной техники, коммуникаций и программного обеспечения. В то же время интенсивное использование информационных технологий в промышленной автоматизации вызвало перераспределение функций между

человеком и аппаратурой, обострило проблему взаимодействия человека-оператора с системой управления [1].

Анализ причин большинства аварий в сложных автоматизированных системах в промышленности и на транспорте показал, что если в 60-х годах XX века ошибка человека являлась первопричиной лишь 20% инцидентов, то уже к 90-м годам XX века доля «человеческого фактора» выросла до 80% (рис. 1) [2]. Постоянное совершенствование технологий, непрерывное повышение надежности электронного оборудования и машин, а также экспоненциальный характер представленных зависимостей позволяет утверждать, что доля «человеческого фактора» будет и далее расти.

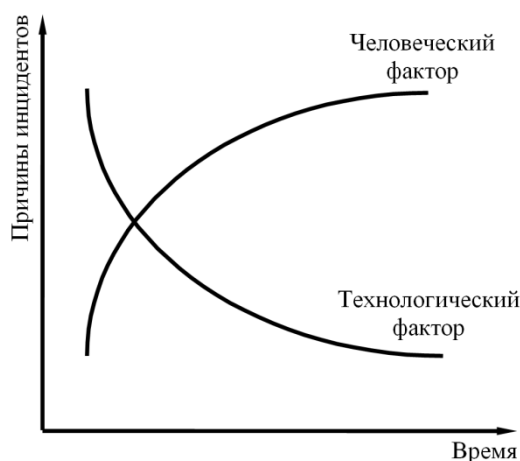


Рис. 1. Причины аварий в сложных автоматизированных системах.

Основной причиной сложившихся тенденций является старый традиционный подход к разработке АСУ, при котором разработчики в первую очередь ориентируются на применение новейших технических и технологических достижений, стремятся повысить степень автоматизации и функциональные возможности системы, недооценивая при этом необходимость реализации эффективного человеко-машинного интерфейса (HMI - Human-Machine Interface). Интерфейса, обеспечивающего взаимодействие человека-оператора с системой управления. Описанная проблема стала особенно актуальна в последние 15 лет, на которые приходится пик внедрения в область промышленной автоматизации инновационных сетевых и компьютерных технологий.

В последние десятилетия в США и Европе проводились интенсивные исследования, посвященные оптимизации и повышению эффективности HMI, результатом которых явился новый подход к разработке и проектированию АСУ. На смену традиционному, «hardware-centered» или «bottom-up, снизу-вверх» принципу проектирования, пришел новый, «human-centered design» или «top-down, сверху-вниз» принцип разработки АСУ. Ярким примером реализации нового подхода явились SCADA-системы, которые иногда также называют SCADA/HMI системами [3].

SCADA (Supervisory Control And Data Acquisition - диспетчерское управление и сбор данных) в настоящее время одна из наиболее интенсивно развивающихся и перспективных технологий разработки программного обеспечения АСУ во многих отраслях промышленности. В наибольшей степени управление на основе SCADA применимо для автоматизации непрерывных и распределенных технологических процессов [3,4].

Целью данной работы является исследование и разработка методов интеграции специального программного обеспечения АСУТП, реализованного на языках программирования высокого уровня, в современные SCADA системы.

Изложение основного материала. Возможны два способа интеграции стороннего программного обеспечения в современные SCADA:

1. Реализация разработанных способов, методов, алгоритмов на одном из поддерживаемых выбранной SCADA системой языков программирования стандарта МЭК 61131 (SFC, IL, ST, LD, FBD) [4];
2. Реализация разработанных алгоритмов на общепринятых языках программирования высокого уровня, с дальнейшим их подключением к SCADA системе в виде внешнего модуля;

Первый вариант влечет за собой значительные трудности, связанные с особенностями программирования алгоритмов на языках стандарта МЭК. Данные языки изначально разрабатывались как средство программирования программируемых логических контроллеров (ПЛК) и представляют собой либо значительно упрощенные и измененные варианты классических языков программирования (ST – Pascal, IL – Assembler), либо специализированные языки, предназначенные для решения специфических задач (LD – язык описания релейных схем). Языки SFC и FBD относятся к так называемым «визуализированным» языкам программирования, и предназначены для использования персоналом, не являющимся специалистами в области программирования. Программы на этих языках создаются как конструктор из уже готовых функциональных блоков, находящихся в специальной базе данных.

Очевидно, что программирование сложных и разветвленных алгоритмов на языках стандарта МЭК связано со значительными трудностями, если вообще возможно.

Второй вариант решения задачи интеграции разработанного программного комплекса в современные SCADA, является более предпочтительным. Он позволяет реализовать программное обеспечение практически на любом современном языке программирования высокого уровня (Delphi, C++, VBA и т.д.), а для интеграции полученных программ в SCADA использовать средства предусмотренные разработчиками выбранной SCADA системы, а также стандартные средства операционной системы.

В процессе интеграции также необходимо решить следующие задачи:

1. Предусмотреть возможность двустороннего обмена информацией между SCADA системой и разработанным программным обеспечением. Типы

данных, используемые в современных языках программирования высокого уровня, могут не иметь аналогов в целевой SCADA системе и наоборот.

2. Реализовать управление запуском и остановом различных модулей разработанного комплекса программ из SCADA системы. Какие-то модули встраиваемого программного обеспечения подлежат непрерывному циклическому пересчету, другие запускаются периодически, либо по мере необходимости.

Решение поставленных задач возможно путем создания специализированной промежуточной программы, обеспечивающей ретрансляцию и согласование типов данных, а также управление комплексом программ, в соответствии с командами, поступающими из SCADA системы.

Предположим, что программное обеспечение, подлежащее интеграции в SCADA, решает задачу оптимального управления неким технологическим процессом по математической модели и состоит из следующих основных программных модулей, написанных на произвольном языке программирования высокого уровня:

1. Модуль математической модели процесса;
2. Модуль идентификации мат. модели;
3. Модуль расчета оптимальных режимов процесса по математической модели.

Необходимо учесть, что различные составляющие данного программного обеспечения работают в разных режимах. Модуль оптимизации функционирует в режиме реального времени, непрерывно рассчитывая и уточняя оптимальные значения заданий для регуляторов, программно реализованных в ПЛК. Модуль идентификации включается и вычисляет новые значения настроечных коэффициентов математической модели периодически, при выходе погрешности модели по целевым расчетным параметрам за установленную границу. Сама математическая модель используется непрерывно в ходе расчета оптимальных режимов и периодически при идентификации.

Обобщенная структура взаимодействия SCADA с разработанным программным обеспечением представлена на рис. 2.

Взаимодействие SCADA и разработанного программного обеспечения происходит через модуль-ретранслятор, который решает задачу двустороннего обмена и преобразования информации, а также запуска и останова отдельных программных модулей в соответствии с управляющими командами, поступающими из SCADA. Модуль-ретранслятор также связан с АРМ оператора, это позволяет осуществлять настройку и модификацию используемого программного обеспечения. Математическая модель взаимодействует со SCADA через блок расчета оптимальных режимов процесса, который работает непрерывно, а также через блок идентификации настроечных коэффициентов модели, который запускается по команде SCADA при потере моделью адекватности.

Представленный в работе метод подключения внешних программных модулей к современным SCADA экспериментально апробирован на примере SCADA системы TRACE-MODE 6 и программного обеспечения для

моделирования и статической оптимизации массообменных, совмещенных и каталитических процессов с распределенными управляющими воздействиями. Данное программное обеспечение было разработано в рамках научно-исследовательских работ (НДР № 0111U008143 «Разработка и исследование компьютерно-интегрированных систем управления (КИСУ)») на кафедре компьютерно-интегрированных технологий и метрологии Украинского государственного химико-технологического университета на языках программирования высокого уровня Delphi и C++.

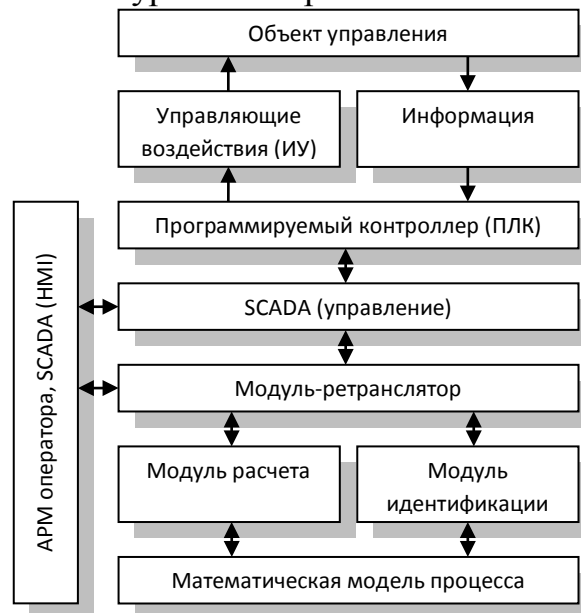


Рис. 2. Структурная схема взаимодействия разработанного программного комплекса со SCADA системой

SCADA система TRACE MODE компании Adastra Ltd получила достаточно широкое распространение на территории нашей страны и выгодно отличается от других известных аналогов невысокой стоимостью без ущерба функциональности, широкой поддержкой различного оборудования, наличием документации и технической поддержки на русском языке.

Основным исполнительным модулем данной системы является монитор реального времени (МРВ), который включает в себя сервер реального времени, графический клиент и реализует следующие основные функции [5]:

- сбор данных с УСО через встроенные и пользовательские драйверы TRACE MODE, DDE и OPC;
- первичная обработка информации - фильтрация, масштабирование, контроль границ и т.д.;
- управление технологическим процессом и регулирование по алгоритмам, запрограммированным на языках стандарта МЭК 61131;
- визуализация информации на мнемосхемах и трендах (HMI);
- ведение отчета тревог;
- предоставление HMI-информации клиентам и другим серверам TRACE MODE;

MPB обладает широкими возможностями для реализации HMI, и поддерживает неограниченное количество УСО, используя различные программные и аппаратные интерфейсы для их подключения.

Рассмотрим принципы подключения внешних программных модулей к SCADA TRACE-MODE, наиболее просто это сделать следующим образом.

Язык программирования Техно-ST, поддерживаемый данной SCADA, позволяет задействовать функции из внешних динамических библиотек (DLL). Библиотеки должны располагаться в директории, из которой запускается монитор реального времени, или в директории, путь к которой указан в переменной PATH операционной системы. Для конфигурирования вызовов внешних функций используются специальные табличные редакторы (рис. 3-5).

Имя подключаемой библиотеки и комментарий к ней указываются в редакторе, который открывается в окне структуры программы, в группе «Внешние библиотеки» (рис. 3). Имя библиотеки указывается без расширения.

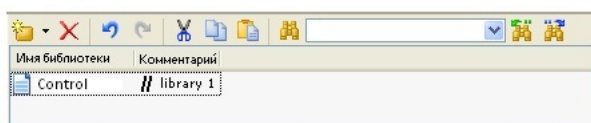


Рис. 3. Редактор имени внешней библиотеки

Библиотечные функции, которые необходимо вызывать из внешней динамической библиотеки описываются в табличном редакторе, который также открывается в окне структуры программы на имени внешней библиотеки (рис. 4).

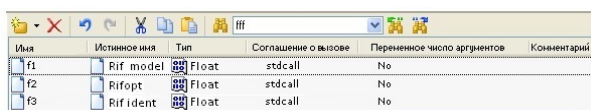


Рис. 4. Редактор функций внешней динамической библиотеки

Аргументы внешних функций описываются в табличном редакторе, который открывается в группе «Аргументы» функции в окне структуры программы (рис. 5).

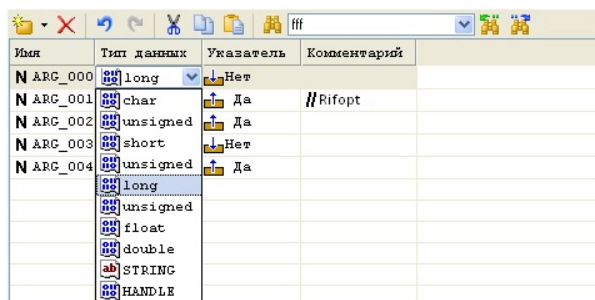


Рис. 5. Редактор аргументов функций внешней динамической библиотеки

Управляющий модуль «Control» (рис. 3) работает в качестве модуля-ретранслятора (рис. 2) и обеспечивает конвертацию типов данных использованных в специальном программном обеспечении и неподдерживаемых на уровне SCADA, таких как динамические переменные и многомерные массивы. Также в данном модуле была реализована система флагов на основе переменных логического типа «Bool», позволяющая управлять работой, запуском и остановом подпрограмм, реализующих математическую модель, алгоритмы идентификации и расчета оптимальных режимов соответствующих технологических процессов по математической модели.

Выводы.

В работе исследована проблема интеграции специального программного обеспечения информационно-управляющих систем в современные SCADA системы. Рассмотрены возможные способы решения поставленной задачи, выявлены их особенности, преимущества и недостатки. Обоснован выбор наилучшего варианта решения задачи и произведена его экспериментальная апробация на примере SCADA TRACE-MODE 6 и специального программного обеспечения для моделирования и статической оптимизации массообменных, совмещенных и каталитических процессов с распределенными управляющими воздействиями.

Список литературы

1. Минитаева А.М. Введение в проблему человеко-машинного интерфейса с учетом взаимодействия вычислительной машины / Современные проблемы информационной безопасности и программной инженерии: сб. избран. стат. науч.-методологич. сем. № 3 кафедры ИБиПИ / А.М. Минитаева // М.: Изд-во «Спутник», 2012. С. 73–77.
2. Расмуссен Й. Чему учат ошибки человека? // Психология труда и организационная психология: современное состояние и перспективы развития. Хрестоматия / Ред. А.Б. Леонова, О.Н. Чернышева. - М., 1995. - С. 117-135.
3. Пономарев О.П. SCADA- системы: уч.-мет. пособие / О.П. Пономарев. – Калинин-град, 2006. – 78 с.
4. Петров И. В. Программируемые контроллеры. Стандартные языки и приемы прикладного проектирования / И. В. Петров; под ред. проф. В. П. Дьяконова – М. : СОЛОН-Пресс, 2004. – 256с.
5. Ефимов И.П. SCADA – система Trace Mode / И. П. Ефимов, Д. Солуянов-Ульяновск : УлГТУ, 2010г. – 158 с.

*Рекомендовано к публикации д.т.н. Мещеряковим Л.И.
Поступила в редакцию 16.01.15*