DOI: 10.17117/na.2015.10.03.058

Поступила (Received): 19.10.2015

<http://ucom.ru/doc/na.2015.10.03.058.pdf>**Валев О.В.****Динамизация поведения алгоритмов
управления в опытных системах****Valev O.V.****Dynamic behavior of the control algorithms in
the experimental and pilot systems**

В настоящей статье рассмотрена актуальность вопроса оперативного изменения поведенческих свойств алгоритмов управления в экспериментальных и опытных системах.

Предложен и обоснован вариант решения указанной проблемы на примере реально разрабатываемого проекта с использованием системы «Trace Mode 6»

Ключевые слова: алгоритм, управление, экспериментальный, SCADA, trace mode

Валев Олег Владимирович*Кандидат технических наук, доцент**Филиал Московского государственного института информационных технологий, радиотехники и электроники**Московская область, г. Сергиев Посад, ул.**Центральная, 10*

In this article we consider the relevance of operational changes behavioral properties of control algorithms in experimental and prototype systems. It was proposed the variant of the decision of this problem on the example of an actual development project using the system "Trace Mode 6"

Key words: algorithm, control, experimental, SCADA, trace mode

Valev Oleg Vladimirovich*Candidate of Engineering Sciences, Associate Professor**Branch of the Moscow state institute of information technologies, radio engineering and electronics**Moscow region, Sergiev Posad, Central st., 10*

Введение. Алгоритм является одним из ключевых понятий в теории вычислений и представляет собой законченный набор правил, необходимых для выполнения некоторой программы или подпрограммы. По мнению профессора Геннадия Михайловича Алакоза, в целом, существует три основных свойства, в соответствие с которыми любую последовательность действий можно классифицировать как алгоритм, это:

– массовость – гарантирует выполнение целого класса однородных и однотипных процедур;

– детерминированность – предполагает однозначное понимание каждой инструкции алгоритма и гарантирует воспроизводимость результата вычислений, проводимых в одинаковых условиях;

– результативность – гарантирует конечность алгоритма, а значит, и времени решения задач.

При решении задач автоматизации выполнение алгоритмов управления происходит циклически на протяжении всего времени мониторинга и управления процессом. В свою очередь, сама исполняющая алгоритм программа, а в большинстве случаев она даже ни одна – их целый набор, работает под управлением некоторой исполнительной среды (по аналогии с CLR [1] платформы «.NET»). Данную исполнительную среду в области разработки программного обеспечения (ПО) для нужд автоматизации часто называют монитором реального времени или просто монитором. Он играет одну из ключевых ролей в раздельной модели построения SCADA-систем (SCADA – supervisory control and data acquisition, диспетчерское управление и сбор данных). Разделение, как минимум, между инструментальной и исполнительной частями лежит в основе работы с любой современной SCADA-системой и системой автоматического управления (CAU), основанной на ней.

Не все объекты, подлежащие автоматизации, так легко этой самой автоматизации поддаются. С точки зрения управления объектами, степень изученности и готовности которых варьируется где-то на уровне экспериментального или опытного образца, имеется ряд задач, требующих внимания. В этом случае, главным образом вопрос встает о том, что до конца не известно поведение подобных объектов даже на этапе испытательных работ. И на протяжении таких испытаний технологическим персоналом вводятся различного рода предложения с поправками в работе ПО CAU.

Внесение поправок – это двусторонний процесс и затрагивает интересы не только технологов, но и разработчиков CAU. Часто поводов для таких поправок так много и возникают они так часто, что казалось бы тривиальные вещи, связанные с изменениями работы условий в алгоритмах и перекомпилированием программного кода, вызывают проблемы и сдвиг сроков сдачи проекта по заключенному договору. И все от того, что сотрудники технологической сферы не обладают должным набором ни знаний, ни навыков для самостоятельного изменения работы программ управления.

Взаимодействие со SCADA. Следуя общим тенденциям импортозамещения, в качестве примера системы диспетчерского управления и сбора данных для демонстрации разрабатываемого проекта CAU была выбрана интегрированная программная среда «Trace Mode 6» от российской компании «АдАстра». Не вдаваясь в особенности работы с ней (об этом более чем подробно можно узнать в [2, 3]) отметим ключевые особенности и термины, которые помогут понять суть решаемой проблемы. Основным понятием «Trace Mode 6» является «канал». Канал – это структурная переменная, подобно той, которая свойственна типу «запись» в языках программирования общего назначения. Каналы делятся на несколько классов и помогают решить четыре следующие главные задачи:

- сбор показаний датчиков от модулей ввода-вывода;
- передача команд управления модулям ввода-вывода;
- вызов программ управления и обработки человеко-машинного интерфейса;

– выполнение запросов для записи, выборки и обновления данных в базе данных (БД).

На последнем моменте из перечисленного списка остановимся подробнее. Связь с БД осуществляется с использованием драйвера ODBC (Open Database Connectivity). И строка доступа, созданная непосредственно в редакторе канала заданного типа, позволяет организовать соединение не только с БД конкретно взятой СУБД, а также с любым документом, поддерживающим указанный способ связи. Таким образом, автором предлагается обеспечить замену абсолютных значений в элементах схемы алгоритма (читай программы) на их относительные величины, подставляемые, например, из документа «Microsoft Excel». Каким образом обеспечить эту подстановку излагается ниже.

При проектировании САУ в «Trace Mode» для решения рассматриваемого вопроса потребуется дополнительно создать следующие каналы:

– Канал класса «CALL» с типом вызова «SQLQuery» для связи с файлом «Microsoft Excel».

– Канал класса «CALL» с типом вызова «ChGroupReq» для импорта значений из документа и их подстановку в программу.

Для каждого алгоритма программы необходимо отвести отдельную колонку в таблице «Excel», ячейки которой должны содержать значения элементов его блок-схемы, подлежащих изменению. Для взаимодействия с драйвером «ODBC» все необходимые колонки упоминаемого документа должны быть объединены в именованную область.

Рис. 1. Схема импорта данных алгоритма «НЭ1..3» в ПО САУ на базе «Trace Mode 6»

**Рис. 2. Блок-схема алгоритма работы НЭ1..3
с изменяемыми значениями ее элементов**

В первом созданном по списку канале каждый вновь введенный в нем аргумент определяет наименование указанной колонки (алгоритма, программы). Он должен быть связан с атрибутом «Реальное значение» второго канала из приведенного списка для того, чтобы обеспечить выгрузку данных из нужной колонки таблицы «Excel» в его аргументы. Таким образом, количество аргументов первого канала, равным счетом как и количество каналов второго вида должно соответствовать требуемому числу задействованных программ. Количество аргументов канала «ChGroupReq» должно соответствовать количеству подстановочных значений в блок-схеме алгоритма, которому он соответствует.

На рисунке 1 приведена схема, обобщающая принцип и порядок действий ПО при загрузке данных алгоритма из таблицы «Excel» на примере управления группой нагревателей «НЭ1..3».

Блок-схема рассматриваемого в качестве примера алгоритма и представление изменяемых в нем данных в таблице «Excel» приведены ниже, на рис. 2.

Разработка человеко-машинного интерфейса. В целом, для изменения поведенческих свойств алгоритмов управления экспериментальных и опытных систем достаточно соблюдать вышеуказанный порядок действий и ограничиться использованием документа «Microsoft Excel» на фоне ПО САУ. Но это не всегда может оказаться удобным с точки зрения уже переложенного на код и требуемого дополнительной отладки алгоритма силами все тех же технологов. В этом случае содержимое документа табличного редактора не дает полного представления о том, какое значение, какому элементу блок-схемы алгоритма соответствует, что влечет за собой необходимость дополнительно держать под рукой описание блок-схем алгоритмов в том или ином виде. Для устранения изложенного недостатка целесообразно использовать ПО – оно же конструктор блок-схем, консолидирующее в себе не только возможности отображения требуемой колонки с подстановочными значениями таблицы «Excel», но и позволяющего непосредственно изображать блок-схемы алгоритмов, предоставлять возможность изменять значения их элементов, а также обеспечивающего быстрый экспорт измененных значений обратно в документ исходного формата.

Таким образом, схема взаимодействия системы «Конструктор блок-схем – файл «Excel» – ПО САУ» может выглядеть следующим образом:

Рис. 3. Схема взаимодействия компонентов в трехзвенной организации системы обеспечения гибкости алгоритмов

ПО для создания и работы с блок-схемами алгоритмов должно предоставлять следующие основные возможности:

- визуальную разработку блок-схем алгоритмов программ управления и тегирование подстановочных значений в элементах блок-схем для их идентификации;
- экспорт подстановочных данных элементов блок-схем в файл формата «Microsoft Excel» заданной структуры;

– изменение подстановочных значений элементов блок-схемы ранее разработанного и реализованного алгоритма и сохранение новых актуальных данных в файле.

В соответствии с перечисленными выше требованиями экранная форма программы-конструктора блок-схем алгоритмов может выглядеть, к примеру, как на рисунке 4.

Рис. 4. Образец экранной формы программы для создания и редактирования блок-схем алгоритмов

Заключение. Как видно из выше изложенного материала, отличительной особенностью предлагаемого решения является очевидная простота реализации. Нет ничего нестандартного. Несмотря на то, что приведенный пример затрагивает использование конкретных программных продуктов, это не ограничивает нас от применения в качестве средств разработки ПО САУ каких-либо других инструментальных средств. Поэтому с большой долей утверждения можно также сказать, что решению в чем-то свойственен и универсальный характер.

Список используемых источников:

1. Microsoft. Среда CLR.
URL: <https://msdn.microsoft.com/ru-ru/library/8bs2ecf4%28v=vs.110%29.aspx>
2. АдАстра. TRACE MODE 6 Интегрированная SCADA/HMI-SOFTLOGIC-MES-EAM-HRM-система для разработки АСУ ТП, АСКУЭ и систем управления производством. Т. 1 и 2. М. 2011.
3. АдАстра. TRACE MODE 6 & T-FACTORY Интегрированная платформа для управления производством. М. 2008.
4. Microsoft. The Environment CLR.
URL: <https://msdn.microsoft.com/ru-ru/library/8bs2ecf4%28v=vs.110%29.aspx>
5. AdAstra. Integrated TRACE MODE 6 SCADA/HMI-SOFTLOGIC-MES-EAM-HRM system for developing automated process control systems, utility metering systems and production management. М. 2011.
6. AdAstra. TRACE MODE 6 & T-FACTORY Integrated platform for production management. М. 2008.

© 2015, Валев О.В.

Динамизация поведения алгоритмов управления в
опытных системах

© 2015, Valev O.V.

Dynamic behavior of the control algorithms in the
experimental and pilot systems