УДК 681.51

М.А. ПОЛЯКОВ
Запорожский национальный технический университет
polyakov@zntu.edu.ua

ЛОГИЧЕСКОЕ УПРАВЛЕНИЕ ОБЪЕКТАМИ ЭЛЕКТРИЧЕСКИХ СИСТЕМ В СРЕДЕ ПРИЛОЖЕНИЯ ЧЕЛОВЕКО-МАШИННОГО ИНТЕРФЕЙСА

Principles of realization of logical control are described in the Man Machine interface application. The hierarchy of control levels and principles of realization of operating and control FSM is offered.

Введение. Системы управления (СУ) сложными объектами электрических систем реализуются, как правило, как распределенные, многоуровневые системы. Уровень управления представлен в них программируемыми логическими контроллерами, а верхний уровень – автоматизированными рабочими местами оператора (диспетчера) (АРМ). Аппаратные средства АРМ – это промышленные компьютеры или панели операторского интерфейса. Программное обеспечение АРМ содержит приложение человеко-машинного интерфейса(MMI - Man Machine Interface или HMI – Human Machine Interface), которое функционирует в среде операционной системы (ОС) вычислительного устройства АРМ. Приложения MMI проектируются и исполняются в среде соответствующих пакетов программ [1]. Например, система мониторинга и управления силового трансформатора [2] содержит АРМ оператора с приложением MMI, которое спроектировано с использованием пакета программ RSView32 [3] компании Rockwell Automation (США).

Приложения MMI оперируют с данными, которые описываются в базе тегов. Теги подразделяются на приборные (поступающие извне через серверы данных) и теги памяти (пользовательские и системные теги, генерируемые приложением MMI). Приложение MMI могут выполнять ряд действий изменяющих значение тегов и, даже состояние других приложений, исполняемых на компьютере приложения MMI или других компьютерах. Действия выполняются как команды присвоения тегам значений выражений, команды приложения, макропоследовательности или программы. В составе пакетов для проектирования приложений MMI есть средства привязки действий к событиям. События это результат оценки данных. К событиям относят истинность выражений содержащих теги, особые оговоренные значения тегов, факт ввода информации оператором (с помощью клавиатуры или манипулятора, например мыши). Таким образом, функциональные возможности программ MMI значительно шире, чем просто ввод и вывод информации из системы управления в форме удобной для восприятия оператором.

Цель исследования. Логическое управление – одна из функций СУ объектами электрических систем. В ходе такого управления воздействия СУ на объект управления (ОУ), в общем случае, зависят от технического состояния ОУ и входных сигналов СУ. Такая зависимость задает определенное поведение ОУ в процессе управления, которое формализуется автоматной моделью [4].

В распределенной, многоуровневой СУ задачи логического управления решаются, в основном, на уровне управления в среде приложений ПЛК. Вместе с тем, ряд задач логического управления выносятся на уровень приложений MMI. Например, управление такими видами поведений в приложениях MMI, как поведение подсистем графики, тревожной подсистемы, регистрации данных и деятельности, защиты и обновления данных в базе тегов. В литературе[1-3] описаны пользовательские аспекты проектирования подсистем приложения MMI. Вместе с тем, в доступной автору литературе, отсутствуют средства и модели описания автоматного, в смысле теории конечных автоматов[4], поведения приложения MMI. Это затрудняет проектирование действий, зависящих от предыстории данных и событий, оценку комплексности и непротиворечивости взаимосвязей событий и действий в проекте и, в конечном итоге, негативно сказывается на качестве приложения MMI.

Цель исследования – раскрыть сущность и способы реализации логического управления в приложении MMI

Основная часть. Предлагается рассматривать приложение MMI как многоуровневую программную систему, элементы практически всех уровней, которой могут быть представлены автоматными моделями. Уровни управления приложения MMI приведены в табл. Ниже рассмотрены уровни 1-4. Предполагается, что уровни 5 и 6 будут рассмотрены в дальнейшем в отдельной работе.

Управление приложениями и процессами компьютера представляет нижний уровень иерархии управлений. Команды ОС реализованы как автоматы [4] и могут исполняться как действия (деятельности) автоматов вышестоящих уровней управления. Например, в пакете RSView32 [3] есть возможность выполнить команду ОС (запустить внешнее приложение, удалить файл и др.) как реакцию на событие.

Таблица - Уровни управления приложения MMI

Номер	Наименование уровня	Элементы
6	Управление самоорганизацией	Продукционная система
5	Управление адаптацией	Управляющие автоматы адаптации
4	Управление операционными автоматами	Управляющие автоматы (УА) приложения MMI,
3	Управление операциями пользователя	Операционные автоматы (ОА) приложения MMI,
2	Управление типовыми операциями	Исполняющая система и команды пакета программ MMI.
1	Управление приложениями и процессами компьютера	ОС компьютера: управляющее ядро и команды

Управление типовыми операциями приложения осуществляется исполняющей системой приложения MMI. Под типовыми операциями будем понимать команды доступные в функциональном меню пакета проектирования приложения MMI. Например, в приложении RSView32[3] команда **Set** *<tag_name> <value>* записывает значение *<value>* в тег *<tag_name>*, команда **Display***<file>* запускает указанный в параметре команды файл графического дисплея, а команды **EventOn** *<file>*, **EventOff** *<file>*, соответственно, начинают и прекращают работу указанного файла событий.

Операционные автоматы уровня управления операциями пользователя выполняют преобразования «входные данные – выходные данные», «входные данные – события» и «события – выходные данные». Подразделим ОА на входные, выходные и промежуточные. Входные ОА (ВхОА) выделяют событие из потока входных данных СУ. Это событие поступает с выхода ВхОА на вход УА уровня управления операционными автоматами, Выходные ОА (ВыхОА) формируют выходные данные СУ на основании значений выходов УА. Промежуточные ОА (ПрОА) преобразовывают входные и(или) промежуточные данные, в другие промежуточные данные.. Их выходы не связаны непосредственно со входами УА.

В тех случаях, когда входные данные интерпретируются УА как события, ВхОА может отсутствовать. Чтобы подчеркнуть, что в последнем случае данные из базы тегов превращаются в событие (вход) УА, введем понятие тривиального ВхОА. Типовые ВыхОА вычисляют значения выходных данных как некоторой функции данных на входе при поступлении управляющего воздействия с выхода УА. В тех случаях, когда выходы УА интерпретируются как выходные данные, ВыхОА может отсутствовать. В этом случае, по аналогии с тривиальным ВхОА, введем понятие тривиального ВыхОА. ПрОА выполняют соответствующее преобразование независимо от состояния УА (неуправляемые ПрОА) или с учетом управляющего воздействия с выхода УА (управляемые ПрОА). При соединении ПрОА между собой и (или) ВхОА, ВыхОА образуется композиция ОА.

Алгоритм преобразования данных в ОА специфицируется в приложении MMI с помощью выражений, производных тегов, событий, команд, макрокоманд или VBA – программ. Элементы выражений - это теги, константы, математические, логические, битовые операторы, операторы отношения, встроенные функции (функции тега, времени, файла, математические) и конструкции условной логики. При этом выражение, содержащее знак «равно» с синтаксисом *<tag_name> =<expression>*, где *< tag _name >* - имя тега, в котором будет храниться результат выражения *<expression>*, вводится как команда [3]. Команду

tagY=expression(tagX1, tagX2, .., tagXN), (1)

где **tagY –**производный тег (тег памяти), а **tagX1, tagX2,.., tagXN** - теги входных или промежуточных данных, от которых зависит значение выражения, будем рассматривать как ОА. Если производный тег имеет аналоговый или строковый тип, то команда (1) специфицирует ПрОА, а если цифровой (дискретный) тип, то ВхОА или ПрОА, в зависимости от вида тегов данных выражения (1).

Для задания ВхОА можно использовать события MMI, которые связывают выполнение команд и макрокоманд (последовательности команд) приложения MMI с истинностью выражения. Включим правую часть выражения (1) в условную часть события, а действием события назначим команду **Set** с параметрами **tagS** – имя тега события и **value** – значение события. Как правило, тег **tagS** является цифровым и событие считается произошедшим, если **tagS** имеет значение «1».

Для задания ВыхОА также можно использовать событие, в условной части которого содержится выражение **expression(tagS1, tagS2, .., tagSN),** где **tagS1, tagS2, .., tagSN** - теги входных событий данного ВыхОА, от которых зависит истинность выражения. Действием события назначим команду **Set** с параметрами **tagS** - имя тега выходных данных ОА и **value** – значение данных. Если ВыхОА имеет более одного тега выходных данных, то соответствующие команды **Set** объединяются в макропоследовательность, имя файла которой записывается в поле действия события.

Для специфицирования ОА со сложными алгоритмами (например, регулятор, решатель систем уравнений и т.п.) или большим числом параметров используются VBA – программы, Эти программы могут запускаться командой пакета MMI. Например, в пакете RSView32 [3] это команда **VbaExec** *<имя_подпрограммы>*, которая используется в макрокоманде, поле действия события или в командной строке. Создание VBA – программы выполняется в интегрированной среде программирования, в которой имеется доступ к элементам объектной модели пакета программ MMI. С помощью этих объектов можно взаимодействовать с проектом MMI из VBA – программы.

Управление операционными автоматами представлено в табл. множеством УА. Пусть конечный автомат задан шестеркой объектов $A=\langle S,X,Y,s_0,\delta,\lambda\rangle$, где S - множество состояний; X - множество входных сигналов; Y - множество выходных сигналов; $s_0 \in S$ - начальное состояние; $\delta: S\times X \to S$ - функция переходов; λ - функция выходов. Для автомата Мили функция $\lambda: S\times X \to Y$, а для автомата Мура $\lambda: S \to Y$. На практике применяется также С-автомат, который имеет две функции λ: $\lambda_1: S_1\times X \to Y_1$ и $\lambda_2: S_2 \to Y_2$, где S_1, S_2 это множества состояний подавтоматов Мили и Мура. Состояние $S_{2i} \subset S_2$ подавтомата Мура определяет значения выходов $Y_i \in Y$, то есть действия автомата в текущем состоянии и подмножество входов $X_i \subset X$, существенное для определения переходов УА из состояния S_{2i} в новые состояния. А состояние $S_{1j} \subset S_1$ подавтомата Мили определяет подмножество входов $X_j \subset X$, существенное для задания значений выходов, а также правило определения значений выходов Y_1 и новых состояний, в зависимости от значений входных сигналов из подмножества X_j. По мнению автора, тот факт, что S_1, S_2 это множества различной природы не отражен в известной литературе по С - автоматам.

В приложении ММІ УА могут быть зависимыми или независимыми, образовывать подавтоматы, иерархию и композиции, исполняться параллельно или последовательно во времени на одном или более компьютерах. Для описания взаимосвязей УА в базу тегов приложения ММІ вводятся дополнительные теги, а в редакторе событий создаются дополнительные события.

Например, пусть $A_{12}=\langle S_2,X_2,Y_2,s_{02},s_{E2},\delta_2,\lambda_2\rangle$ - подавтомат автомата $A_1=\langle S_1,X_1,Y_1,s_{01},\delta_1,\lambda_1\rangle$, причем s_{E2} - конечное состояние, а S_2,δ_2,λ_2 - не пустые подмножества соответствующих множеств S_1,δ_1,λ_1. В результате выделения подавтомата, исходный автомат A_1 может быть представлен автоматом $A_3=\langle S_3,X_3,Y_3,s_{03},\delta_3,\lambda_3\rangle$, у которого $s_{03}=s_{01}$, $S_3=\{S_1\setminus S_2, S_{12}\}$, $Y_3=\{Y_1\setminus Y_2, Y_{12}\}$, где S_{12} - состояние в котором выполняется подавтомат A_{12}, Y_{12} - выход вызова подавтомата A_{12} в состоянии S_{12}. Причем события, приводящие к переходу из состояния S_{12} в другие состояния блокируются до достижения подавтоматом A_{12} конечного состояния s_{E2}. То есть, автоматы A_3 и A_{12} исполняются последовательно во времени. Для задания такой структуры автоматов в приложении ММІ в базу тегов дополнительно к тегам автомата A_1 внесем следующие теги: «s_{E2} - конечное состояние подавтомата A_{12}», «Y_{12} - выход вызова подавтомата A_{12}». Кроме того, в перечень событий вносятся события: «запуск подавтомата A_{12}», «подавтомат A_{12} достиг конечного состояния s_{E2}», а при описании событий переходов автомата A_3 в условие срабатывания перехода входит проверка отсутствия активности подавтомата A_{12}.

Иерархию УА опишем отношением «руководства-подчиненности», в соответствии с которой, вышестоящий в иерархии автомат руководит (управляет) нижестоящим, а последний исполняет операции управления. Классифицируем операции управления по тем элементам управляемого автомата, которые они затрагивают:

1. Управление входами X - это блокировка/разрешение всех или части входов из множества X. Очевидно, что блокировка всех входов приведет к остановке, «замораживанию» автомата в некотором состоянии, а блокировка части входов приведет к изменению функций δ и λ
2. Управление выходами Y - это блокировка/разрешение всех или части выходов из множества Y, выходов определенного состояния из множества S.
3. Управление состояниями S - это сброс автомата в начальное состояние, переход в конечное или в произвольное заданное промежуточное состояние.
4. Управление функциями δ, λ - это загрузка новой функции; изменение параметров функции путем блокировки/разрешения всех или части переходов задаваемых отображением $\delta: S\times X \to S$ или выходов задаваемых отображениями $\lambda: S\times X \to Y$, $\lambda: S \to Y$ в некотором «максимальном» отображении. Управление путем изменения параметров функции, в отличие от управления входами и выходами, требует предварительного задания этого максимального отображения. В результате загрузки новой функции можно изменить тип автомата, т.е. заменить автомат Мили автоматом Мура и наоборот.
5. Управление синхронизацией - это задание вида синхронизации (синхронный, асинхронный) автомата. Для синхронных автоматов управление дополнительно включает разрешение/блокировку входа синхронизации, задание длительности такта и фаз сигналов синхронизации, способа их привязки к системному

времени, коррекцию времени. Очевидно, что блокировка сигналов синхронизации эквивалентна блокировке всех входов автомата. Управление синхронизацией исполнения нескольких автоматов это задача вышестоящего УА, которая сводится к разрешению/запрету параллельной работы управляемых автоматов в текущий момент времени, изменении уровня приоритета исполнения автоматов.

Управление автоматами по описанной выше схеме будем называть внешним управлением. В том случае, когда элементы, задающие автомат изменяются по «инициативе» самого автомата будем говорить о самоуправлении. Целью самоуправления является повышение качества операций автомата i - го уровня по управлению автоматом $(i-1)$ – го уровня. Самоуправление реализуется разделением УА i - го уровня на, собственно, УА и ОА определения функции качества управления (ОА ФКУ). Множество входов ОА ФКУ определим, как $X_{\text{ФКУ}} = X_{\text{УА}} \cup Y_{\text{УА}} \cup S_{\text{УА}}$, а множество выходов $Y_{\text{ФКУ}}$ - как систему действий изменяющих элементы УА. Выходы $Y_{\text{ФКУ}}$ соединены с входами управления автомата УА. Активность выхода $Y_{\text{ФКУ}}$ возникает в результате идентификации ОА ФКУ определенных событий в управляемом автомате. В качестве примера таких событий отметим события статистики и контроля поведения УА. Событиями статистики могут служить, например, превышение заданного числа заданных событий из множества $X_{\text{УА}}$, количество вхождений в определенное состояние исходного УА, превышение заданного времени нахождения УА в некотором состоянии и т.п.. Примеры событий контроля поведения УА - это распознавание заданной цепочки (последовательности) значений входов и(или) выходов УА, распознавание несоответствия фактических функций выходов и переходов УА некоторому эталонному поведению.

Проектирование УА в среде проектирования приложения MMI включает [5]:

1. По возможности, разбиение уровня управления ОА на независимые УА. Выбор принципов синхронизации и последовательности исполнения этих УА.
2. По возможности, структурирование независимого УА, т. е. выделение в нем иерархических уровней и параллельных УА. Выбор принципов синхронизации и последовательности исполнения этих УА.
3. Внесение в базу тегов приложения MMI тегов состояний и взаимосвязей УА по иерархии и последовательности выполнения. Затем проектирование функций выходов λ и переходов δ для каждого УА с использованием редактора событий.
4. Проектирование средств визуализации автоматного поведения приложения MMI - экранов подсистемы графики с изображением графов переходов УА, в которых активные, в данный момент, состояния и переходы выделяется средствами анимации графических объектов.
5. Верификация структуры автоматов приложения, в том числе поиск паразитных автоматов, образованных обратными связями выходов автоматов с его входами через ОУ или другие, внешние по отношению к автоматам приложения MMI, автоматы.

Заключение. Таким образом, верхний уровень системы логического управления объектами электрических систем может быть реализован как приложение MMI, а это приложение представлено как многоуровневая система взаимодействующих автоматов. Автоматы могут конфигурироваться такими средствами пакетов MMI, как редакторы тегов, событий и др. или программироваться как VBA - программы. Переход от конфигурирования отдельных событий СУ к проектированию взаимосвязанных автоматов позволяет обеспечить полноту и непротиворечивость автоматного поведения, повысить функциональную гибкость компонентов СУ. В дальнейшем предполагается более детально исследовать уровни управления адаптацией и самоорганизацией приложения MMI СУ.

Список литературы

1. Системы диспетчерского управления и сбора данных (SCADA – системы). - Мир компьютерной автоматизации, №3, 1999, с.6-9.

2. Рассальский А.Н. Система мониторинга и управления силовых трансформаторов.- Електротехніка і Електромеханіка. 2005, №2.

3. RSView32. Руководство пользователя. Компания Rockwell Automation. Публикация. 9399-2SE32UG-FEB 97.

4. Карпов Ю.Г. Теория автоматов. СПб.: Питер,2002.-224 с.: ил..

5. Поляков М.А. Реализация автоматного поведения в приложении визуализации контроллерной системы управления. Международная научно-техническая конференция «Автоматизация: проблемы, идеи, решения», 8-12 сентября 2008 г. [Текст]: [Материалы]/ редкол.: В.Я. Копп [и др.]/- Севастополь: Изд-во СевНТУ, 2008.-296 с.; 21см.-100 экз.- ISBN 978-966-2960-32-7. с.239-240.

Надійшла до редколегії 08.05.2009 *Рецензент: І.П.Заболотний*

М.А. ПОЛЯКОВ
Запорожский национальный технический университет

Логическое управление объектами электрических систем в среде приложения человеко-машинного интерфейса. В статье представлено получение зависимости задающей определенное поведение объекта управления в процессе управления, которое формализуется автоматной моделью

Логическое управление, объект, электрическая система, человеко-машинный интерфейс

М.А. ПОЛЯКОВ
Запорізький національний технічний університет

Логічне управління об'єктами електричних систем в середовищі прикладення людина–машинного інтерфейсу. В статті подано отримання залежності, яка задає певну поведінку об'єкта управління в процесі управління, що формалізується автоматною моделлю.

Логічне управління, об'єкт, електрична система, людина-машинний інтерфейс