

УДК 519.683.8, 004.4'236

В.М. Дмитриев, Т.В. Ганджа, А.С. Букреев

Моделирование сценариев управления динамическими объектами на основе графического языка X-Robot

С ростом числа функций динамических объектов усложняются алгоритмы, на основе которых функционируют их устройства управления. Также увеличиваются программы, написанные на низкоуровневых языках программирования контроллеров, что усложняет процесс их отладки. Актуальность разработки новых средств формирования и моделирования сценариев функционирования контроллеров обусловлена необходимостью учёта многих наблюдаемых переменных и использования ряда управляющих воздействий, а также нетривиальностью алгоритмов функционирования контроллеров.

Рассматриваются принципы моделирования сценариев управления динамическими объектами с использованием графических нотаций языка X-Robot. Помимо традиционного текстового представления программы, называемой сценарием, для данного языка предложена и разработана графическая интерпретация, ставящая для каждой его команды определенный компонент. Графическое формирование сценариев осуществляется в среде компьютерного моделирования MAPC на логическом уровне многоуровневой компьютерной модели, на объектном уровне которой располагается модель управляемого динамического объекта с включенными в нее моделями исполнительных и измерительных устройств. Это открывает возможности формирования сценариев управления в графической форме и их предварительной отладки на модели динамического объекта. На визуальном уровне многоуровневой компьютерной модели располагаются управляющие компоненты, с помощью которых пользователь имеет возможность воздействовать на модель объекта и модель сценария, а также компоненты-визуализаторы, осуществляющие отображение для пользователя данных для визуализации, которыми могут быть как значения наблюдаемых переменных объекта, так и их обобщенные параметры-функционалы. В настоящее время разработанный графический язык моделирования сценариев адаптирован к контроллеру X-Mega, но ведутся исследования по его развитию и применению к управляющим контроллерам других типов.

Ключевые слова: компонент, контроллер, сценарий, динамический объект, многоуровневая компьютерная модель.

doi: 10.21293/1818-0442-2018-21-2-75-82

В настоящее время при реализации устройств управления получили широкое распространение различные контроллеры: X-Mega [1, 2], Arduino [3] и т.п. Их функционирование основано на сценариях, которые формируются на различных языках программирования. Большинство из них являются или приближены к языкам программирования для логических контроллеров. К большинству из существующих контроллеров созданы программно-инструментальные средства программирования и формирования сценариев в текстовой или графической форме [4]. Зачастую отладка и настройка устройств управления с использованием реального объекта затруднены или невозможны вследствие различных причин. К ним относятся сложность технической реализации исполнительных и измерительных устройств, необходимость расчета или подбора значений их параметров, безопасность использования объекта управления, а также большие финансовые затраты для постановки реальных экспериментов.

Большинство языков программирования контроллеров согласуются с международным стандартом IEC 61131 [5], включающим в себя три графических и два текстовых языка. Графические языки базируются на различных графических платформах: язык LD является программной реализацией электрических схем на базе электромагнитных реле. Язык FD представляет сценарий функционирования контроллера в виде некоторой подпрограммы, а язык SFC основан на базе математического аппарата се-

тей Петри [6]. Текстовые языки аналогичны известным языкам программирования: сценарии функционирования контроллеров, написанные на язык ST, напоминают программы языка Паскаль, а сценарии языка IL подобны программам, написанным на языке Ассемблер.

Для программирования контроллеров с использованием обозначенных языков используются системы CodeSys [7, 8] и ISaGRAF [9]. Автоматизируя все необходимые функции по разработке сценария, прошивки его в контроллер с последующим тестированием и отладкой разработанного кода, данные системы не включают в свой состав средств моделирования управляемых технических объектов. Это не позволяет осуществлять предварительную отладку разработанного сценария на модели объекта, а предполагают работать напрямую с реальным объектом. Использование моделей управляемых объектов направлено на существенное облегчение разработки и отладки сценариев функционирования контроллеров.

Для решения данных проблем в работе предлагается использовать графический язык формирования сценариев управляющих контроллеров X-Robot [10], а также встроенный в среду многоуровневого компьютерного моделирования MAPC [11] модуль графического моделирования сценариев [12]. Язык X-Robot имеет текстовую нотацию, приближенную к низкоуровневому языку Ассемблер, которая не позволяет интегрировать сценарий с моделью объекта, представленной в среде моделирования MAPC. Для

решения проблемы в работе предлагается графическая интерпретация языка X-Robot, позволяющая интегрировать разрабатываемый сценарий с моделью управляемого объекта. Она позволяет разрабатывать сценарии в компонентной форме и осуществлять их предварительную отладку на компьютерных моделях объектов управления, исследуя при этом наблюдаемые характеристики модели управляемого объекта и варьируя различные её параметры, а также управляющие и возмущающие воздействия.

Модуль моделирования сценариев лежит в основе разработки курса лабораторных работ по дисциплине «Основы проектирования систем и средств управления», являющейся базовой профилирующей дисциплиной при подготовке бакалавров по направлению «Системный анализ и управление».

Сценарий языка управления динамическими объектами X-Robot

Поведение динамического объекта описывается сценарием на языке X-Robot [10]. Он разбит на блоки, определяющие отдельные процессы, выполняющиеся параллельно. Каждый процесс имеет собственную локальную ячейку памяти – аккумулятор, собственный таймер и регистр состояния для условного ветвления. К атрибутам процесса относятся флаги ожидания, которые при запуске сценария сбрасываются для всех процессов. Каждый процесс представляет собой безусловный цикл с опционной секцией предварительной настройки, расположенной в начале процесса. За ней идет его рабочая секция, после выполнения последней инструкции которой осуществляется безусловный переход на первую инструкцию рабочей секции. Перед этим производится попытка переключиться на следующий процесс.

При первом запуске сценария последовательно запускаются все процессы, начиная с первого. Если

в тексте выполняемого процесса встречается инструкция ожидания или любого изменения порядка выполнения инструкций, выполняется попытка переключения на следующий процесс. Если его нет, то снова выполняется первый процесс. Инструкции ожидания взводят флаг ожидания процесса. Он перестает обрабатываться диспетчером и не занимает процессорного времени контроллера.

В настоящее время существует два способа формирования сценариев языка X-Robot: текстовый, когда сценарий формируется в любом текстовом редакторе в виде инструкций с их разделением на процессы, и графический, при котором каждой инструкции языка ставится в соответствие компонент, реализующий его в рамках модуля моделирования сценариев среды многоуровневого компьютерного моделирования.

Многоуровневая структура компьютерной модели в модуле моделирования сценариев

Модуль моделирования сценариев в формате языка X-Robot реализован в рамках среды многоуровневого компьютерного моделирования MAPS [13]. Его основными задачами являются:

- моделирование и анализ поведения объекта управления при формировании и выполнении различных команд управления;
- формирование сценариев функционирования управляющих контроллеров и отладка их параллельных потоков на компьютерных моделях объектов управления;
- интерпретация сформированного графического сценария в код с последующим программированием контроллера;
- отладка поведения модели объекта с использованием виртуальных приборов, разработанных в системе виртуальных инструментов и приборов [14, 15].

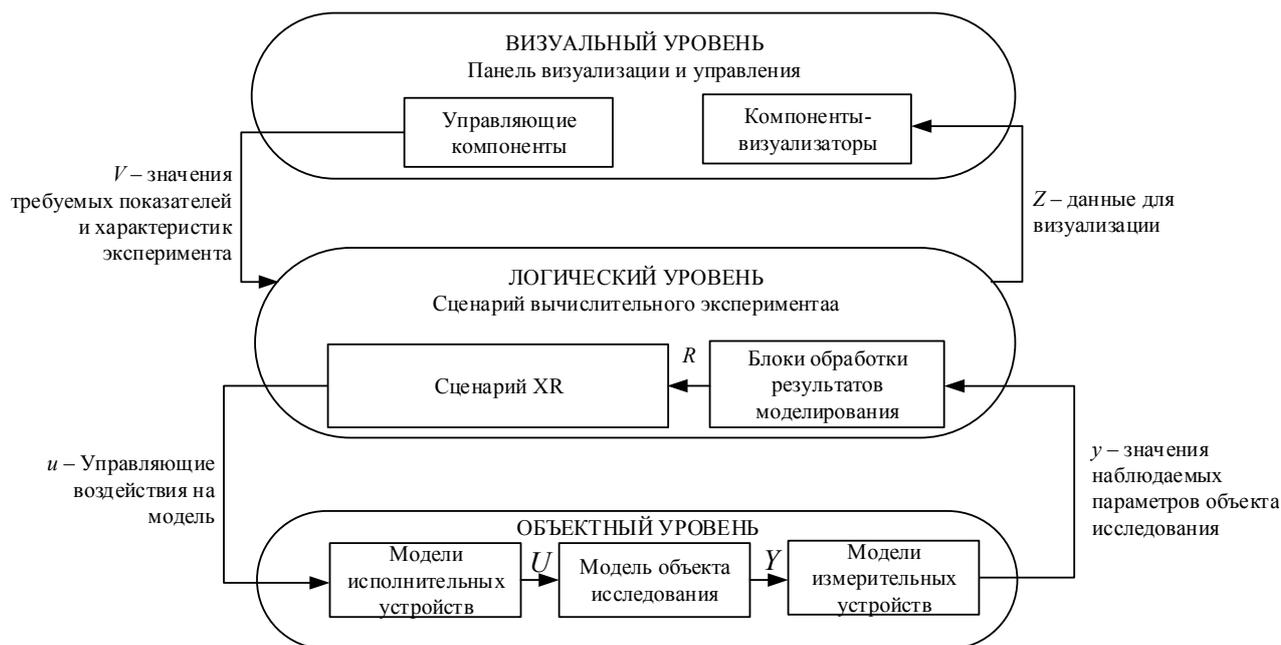


Рис. 1. Структура многоуровневой компьютерной модели для формирования и отладки графических сценариев

Компьютерная модель, предназначенная для формирования и отладки сценариев управления, представлена многоуровневой компьютерной моделью [16] (рис. 1), на трех уровнях которой располагаются взаимосвязанные модели:

- на объектном уровне расположена компьютерная модель объекта управления с подключенными к ней моделями измерительных и исполнительных устройств. Они формируются на основе метода компонентных цепей [17] и в общем случае представляют собой системы алгебро-дифференциальных уравнений. Формирование общей модели в виде системы алгебро-дифференциальных уравнений, линеаризация нелинейных и алгебраизация дифференциальных уравнений с последующим решением системы линейных алгебраических уравнений на каждом шаге анализа по времени производится универсальным вычислительным ядром [18]. С помощью моделей измерительных устройств, имеющих свои графические отображения одновременно на многоуровневой компьютерной модели осуществляется передача значений наблюдаемых переменных объекта управления y с объектного уровня на логический;

- на логическом уровне формируется функциональная модель устройства управления в виде параллельных потоков сценария, взаимосвязанного с моделью объекта управления. Функционирование сценария осуществляется алгоритмом передачи сообщений [19];

- на визуальном уровне располагаются средства визуализации результатов моделирования и работы сценария управления, а также средства интерактивного управления задающими воздействиями [20, 21].

Формируемый на логическом уровне многоуровневой компьютерной модели сценарий функционирования управляющего контроллера в формате графического представления нотаций языка X-Robot на основе значений наблюдаемых переменных объекта управления y осуществляет выработку управляющих воздействий на модель исполнительных устройств u . С формальной точки зрения сценарий управления может быть представлен моделью

$$U = F_{SC}(y),$$

где F_{SC} – функция сценария, заданная в явной форме в виде набора последовательно соединенных элементарных инструкций (команд) языка управления механизмами X-Robot.

Классификация компонентов графического сценария

Компонент, отображающий одну из инструкций (команд) языка X-Robot, представлен на рис. 2 и содержит следующие связи:

- 1) входные информационные связи $X1$ и $X2$, по которым в компонент передаются значения вступающих в операцию операндов;

- 2) управляющие связи $U1$ и $U2$, причем $U1$ предназначена для приема сигнала начала выполнения операции, а $U2$ – для передачи управляющего сигнала следующему компоненту;

- 3) выходная информационная связь Y , по которой передается результат выполнения операции.

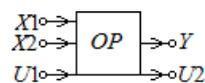


Рис. 2. Обобщенный компонент операции языка X-Robot

В ряде компонентов, описывающих операции ветвления, имеется одна входная управляющая связь $U1$ и одна входная информационная связь $X1$. На основе анализа поступающего по ней значения переменной осуществляется ветвление сценария по одной из выходных управляющих связей: $U2, U3, \dots, Un$. Некоторые компоненты могут иметь только управляющие сигналы.

Рассмотрим основные классы компонентов, из которых осуществляется формирование графических сценариев.

Компоненты начала и окончания процессов и подпрограмм предназначены для обозначения определенных блоков сценария, а также позволяют отделить блок инициализации от основного потока, циклически выполняющегося при работе сценария. Каждый компонент данного класса включает только одну входную $U1$ и одну выходную $U2$ управляющие связи. В данный класс компонентов входят компоненты, обозначающие начало (Begin Process) и конец процесса (End Process), начала цикла процесса (Begin), а также компоненты, при выполнении которых производится безусловный переход на начало процесса как с выполнением блока инициализации (INT), так и без выполнения этого блока (RST).

Класс компонентов взаимодействия сценария с датчиками и исполнителями содержит 8 команд считывания информации с датчиков ($In0, \dots, In7$), а также 8 команд передачи команд исполнителям ($Ex0, \dots, Ex7$). Каждый датчик и исполнитель задаются своим номером от 0 до 255. По входной управляющей связи $U1$ осуществляется передача команды на выполнение, управляющая связь $U2$ предназначена для передачи другим компонентам сигнала о завершении выполнения команды.

Класс компонентов команд ожидания предназначен для остановки некоторого процесса до выполнения определенного события или достижения определенного времени. Обобщенный компонент этого класса представлен на рис. 3.

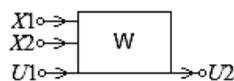


Рис. 3. Обобщенный вид компонентов ожидания

Ожидание начинается с момента времени, когда поступает сообщение на управляющую связь $U1$. В момент наступления ожидаемого события осуществляется передача сообщения по управляющей связи $U2$. По информационным связям $X1$ и $X2$ в компонент при отладке сценария поступают данные, характеризующие значения определенных переменных или текущего времени. К данному классу относятся

компоненты: ожидания заданного промежутка времени (WT), ожидания сообщения (WM), ожидания равенства переменных $X1 = X2$ (WD), ожидания неравенства переменных $X1 \neq X2$ (WND), ожидания больше $X1 > X2$ (WDH), ожидания меньше $X1 < X2$ (WDL). Последовательное соединение данных компонентов в группы представляет собой структуру «ИЛИ». Для их включения в схему «И» используется компонент, обозначающий команду BAND языка X-Robot.

Класс компонентов работы с подпрограммами содержит команды, позволяющие представить один из процессов сценария в виде подпрограммы и вызвать его из другого процесса. Для этого компонентами представлены следующие команды:

– команда вызова подпрограммы (CALL) – устанавливается в процесс, который должен вызвать другой процесс;

– команда ожидания вызова подпрограммы (WR) – устанавливается в процесс-подпрограмму, которая должна выполняться по соответствующей команде CALL.

Класс компонентов арифметических и побитовых операций включает в себя компоненты, для выполнения основных бинарных арифметических и побитовых операций. К ним относятся: присваивание переменной некоторого значения (MOV); суммирование (ADD), вычитание (SUB), побитовое И (AND), побитовое ИЛИ (OR), побитовое исключающее ИЛИ (XOR), команда перестановки байтов (SWAP) и команды побитового сдвига влево (LSL) и вправо (LSR).

Каждая из этих команд выполняется при поступлении на управляющий вход $U1$ компонента (см. рис. 2) сигнала о выполнении над последними полученными значениями операндов $X1$ и $X2$.

Все переменные являются глобальными, и их значения одинаковы для всех потоков сценария. При этом у каждого из процессов есть одна локальная переменная, называемая *аккумулятором*, значение которой для каждого из процессов свое.

Класс компонентов изменения порядка выполнения процесса предназначен для осуществления изменения последовательного порядка выполнения команд процесса, ветвления, циклов и т.п. К этому классу относятся следующие компоненты:

– Компонент «Сравнение» (CMP) при получении сообщений о выполнении по управляющей связи $U1$ осуществляет сравнение последних значений переменных, поступивших по информационным связям $X1$ и $X2$, и на выходную информационную связь Y передает 0, если $X1 = X2$, 1, если $X1 > X2$, и 2, когда $X1 < X2$.

– Компонент «Ветвление» (CASE), имеющий несколько выходных управляющих связей ($U2, U3, \dots, Un$), производит анализ значения, поступившего по информационной связи $X1$, в соответствии с которым посылает сообщение по одной из выходных управляющих связей.

– Компонент «Выбор по сравнению» (CASES) по своей входной информационной связи $X1$ осуще-

ствяет прием значения результата операций сравнения и перенаправление выполнения сценария по одной из ветвей, каждая из которых соответствует ситуациям «Меньше», «Равно» или «Больше».

– Компонент «Возврат из структуры сравнения» (RETS) устанавливается в одну из ветвей, образуемых компонентом «Выбор по сравнению». При получении сообщения по входной управляющей связи данный компонент осуществляет безусловный переход на начало команды CASES, в структуре которой он установлен.

– Компонент «Конец структуры выбора» (ENDC) устанавливается в конец каждой ветви сценария, образованной структурами компонентов CASE и CASES.

Таким образом, описаны все классы компонентов, предназначенных для формирования сценариев функционирования управляющих контроллеров на графическом языке X-Robot.

Сценарий управления дискретным объектом

Светофор, организующий регулирование дорожным движением, является дискретным устройством. Модель светофора, содержащая три фонаря, соответственно красного $L1$, жёлтого $L2$ и зелёного цвета $L3$, загорание которых осуществляется с помощью ключей $S1, S2$ и $S3$, реализована на объектном уровне многоуровневой компьютерной модели и представлена на рис. 4. Регулировочным резистором $R1$ осуществляется регулировка интенсивности свечения фонарей в зависимости от времени суток, погодных условий и других факторов.

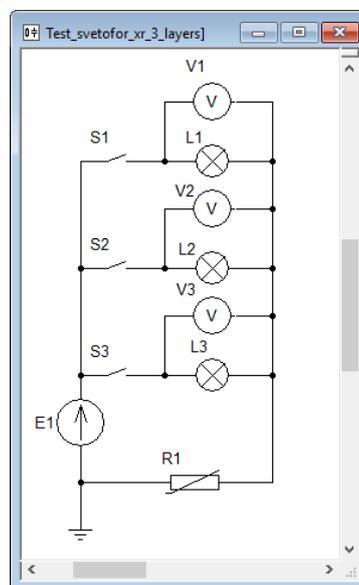


Рис. 4. Компьютерная модель светофора на объектном уровне многоуровневой компьютерной модели

Пусть фонарём красного цвета управляет исполнитель с номером \$08, фонарём жёлтого цвета – исполнитель \$09, а фонарём зелёного цвета – исполнитель \$0a. Каждый из них зажигает определенный фонарь с помощью команды Ex3 подачей максимально возможного значения \$7FFF = 32767. Фонарь погасает подачей с помощью команды Ex3 значения 0.

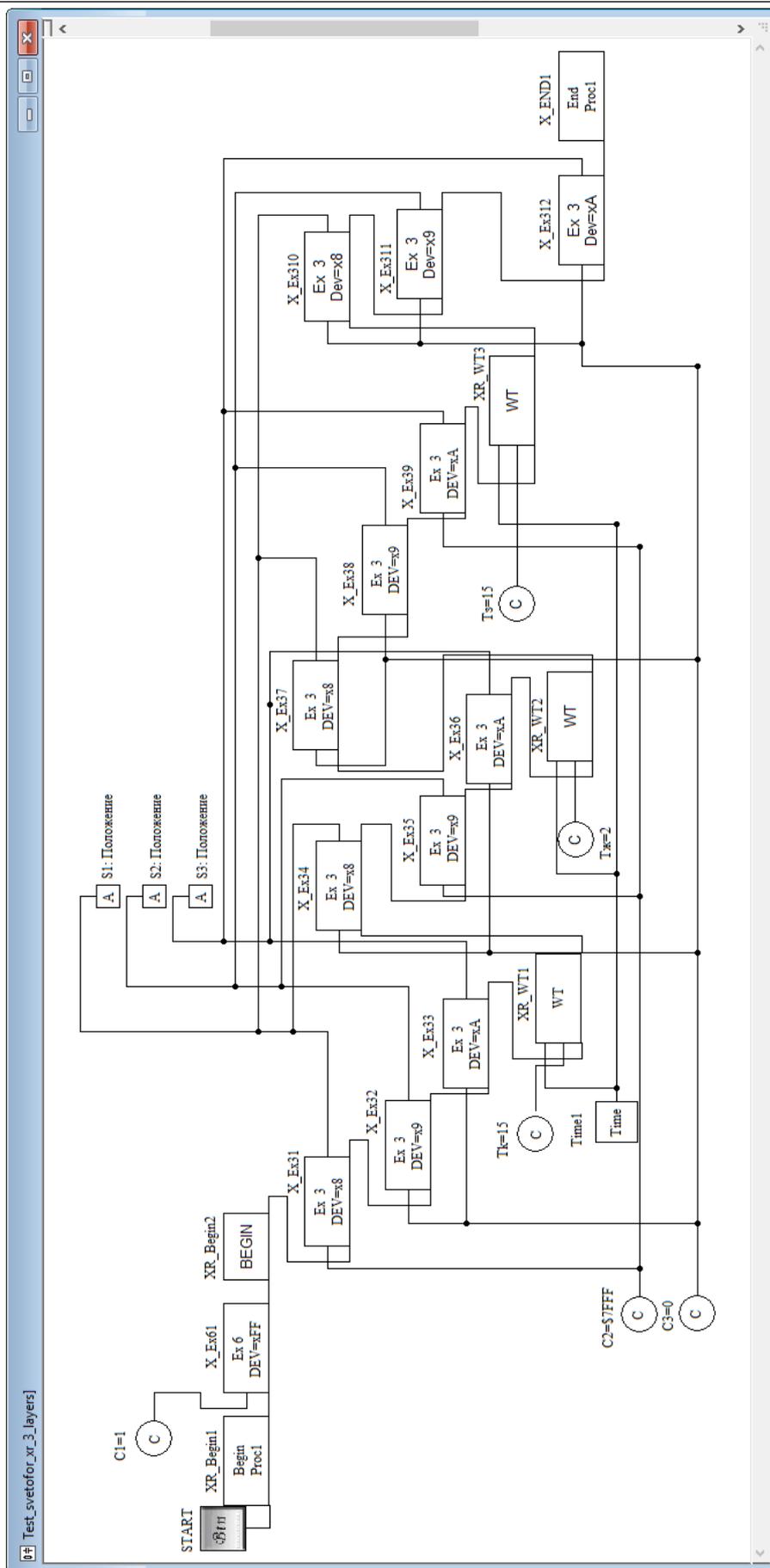


Рис. 5. Графический сценарий управления состоянием фонарями светофора

Инициализация описанных исполнителей, осуществляющих изменение состояний ключей, производится командой Ex6, выполненной для исполнителя инициализации \$ff = 255 с аргументом 1. В этом случае осуществляется инициализация ШИМ-генераторов. Блок инициализации ограничивается от основного цикла процесса компонентом XR_Begin2. С помощью компонентов XR_WT1, XR_WT2 и XR_WT3 производится задержка процесса на определенное время, которое должен гореть соответствующий фонарь светофора.

Графическая форма сценария управления состоянием светофора представлена на рис. 5. На основе представленного сценария будет сформирован код для программирования контроллера, осуществляющего управление состоянием светофора.

Заключение

Модуль моделирования сценариев, реализованный в рамках среды многоуровневого компьютерного моделирования, открывает возможности формирования графических сценариев управления динамическими объектами с их отладкой на компьютерных моделях объектов с включенными в них моделями исполнительных и измерительных устройств. Совокупность графического сценария и модели объекта управления также позволяют производить выбор значений параметров компонентов, удовлетворяющих соответствующим критериям качества управления. Применение разработанных программно-инструментальных средств в процессе обучения направлено на автоматизацию проведения практических и лабораторных занятий, в ходе которых студенты и школьники старших классов приобретают первоначальные знания и умения программирования контроллеров. Данные разработки также будут интересны центрам научно-технического творчества молодёжи, деятельность которых направлена на разработку управляемых динамических объектов.

Работа выполнена при финансовой поддержке Минобрнауки в рамках проекта RFMEFI57717X0266.

Литература

1. Гумеров Р.И. Практикум по микропроцессорам. Часть первая: Микроконтроллеры AVR: руководство. – Казань: КГУ, 2009. – 37 с.
2. Евстифеев А.В. Микроконтроллеры AVR семейств Т1ру и Мега фирмы АТМЕL. – 3-е изд. – М.: Изд. дом «Додэка-XXI», 2006. – 560 с.
3. Петин В.А. Проекты с использованием контроллера Arduino. – СПб.: БХВ-Петербург, 2014. – 400 с.
4. Деменков Н.П. Языки программирования промышленных контроллеров: учеб. пособие / под ред. К.А. Пупкова. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2004. – 172 с.
5. IEC 61131-3:2013 | IEC Webstore | water automation, water management, smart city [Электронный ресурс]. – Режим доступа: <https://webstore.iec.ch/publication/4552>, свободный (дата обращения: 08.05.2018).
6. Мараховский В.Б. Моделирование параллельных процессов. Сети Петри. Курс для системных архитекто-

ров, программистов, системных аналитиков, проектировщиков сложных систем управления / В.Б. Мараховский, Л.Я. Розенблюм, А.В. Яковлев. – СПб.: Профессиональная литература, АйТи-Подготовка, 2014. – 400 с.

7. Петров И.В. Программируемые контроллеры. Стандартные языки и приёмы прикладного программирования / под ред. В.П. Дьяконова. – М.: СОЛОН-Пресс, 2004. – 256 с.

8. Home – CODESYS [Электронный ресурс]. – Режим доступа: <https://www.codesys.com/>, свободный (дата обращения: 03.05.2018).

9. ISaGRAF – leading IEC 61131 and IEC 61499 software [Электронный ресурс]. – Режим доступа: <http://www.isagraf.com/index.htm>, свободный (дата обращения: 03.05.2018).

10. Мальцев Ю.И. Язык управления механизмами X-Robot // Электронные средства и системы управления. – Томск: ТУСУР, 2013. – № 2. – С. 114–118.

11. MAPC – среда моделирования технических устройств и систем / В.М. Дмитриев, А.В. Шутенков, Т.Н. Зайченко, Т.В. Ганджа. – Томск: В-Спектр, 2011. – 278 с.

12. Дмитриев В.М. Среда моделирования сценариев для прототипирования контроллеров / В.М. Дмитриев, Т.В. Ганджа, А.С. Букреев // Научная сессия ТУСУР–2017: матер. Междунар. науч.-техн. конф. студентов, аспирантов и молодых ученых, посвященной 55-летию ТУСУРа, Томск: 10–12 мая 2017 г.: в 8 ч. – Томск: В-Спектр, 2017. – Ч. 4. – С. 107–110.

13. Среда многоуровневого компьютерного моделирования химико-технологических систем / В.М. Дмитриев, Т.В. Ганджа. – Томск: Изд-во Том. ун-та, 2017. – 332 с.

14. СВИП – система виртуальных инструментов и приборов / В.М. Дмитриев, Т.В. Ганджа, В.В. Ганджа, Ю.И. Мальцев. – Томск: В-Спектр, 2014. – 216 с.

15. Дмитриев В.М. Система визуализации и управления вычислительным экспериментом в среде многоуровневого моделирования MAPC / В.М. Дмитриев, Т.В. Ганджа, Т.Ю. Коротина // Доклады ТУСУР. – 2010. – № 1 (21), ч. 2. – С. 149–155.

16. Дмитриев В.М. Принцип формирования многоуровневых компьютерных моделей SCADA-систем для управления сложными технологическими объектами / В.М. Дмитриев, Т.В. Ганджа // Информатика и системы управления. – 2013. – № 2 (36). – С. 24–35.

17. Автоматизация моделирования промышленных роботов / В.М. Дмитриев, Л.А. Арайс, А.В. Шутенков. – М.: Машиностроение, 1995. – 304 с.

18. Дмитриев В.М. Архитектура универсального вычислительного ядра для реализации виртуальных лабораторий / В.М. Дмитриев, А.В. Шутенков, Т.В. Ганджа // Приборы и системы. Управление. Контроль. Диагностики. – 2004. – № 2. – С. 24–28.

19. Григорьева Т.Е. Дискретно-событийное моделирование в СМ MAPC для курса «Системы массового обслуживания» // Доклады ТУСУР. – 2014. – № 1 (31). – С. 152–155.

20. Дмитриев В.М. Компьютерное моделирование визуальных интерфейсов виртуальных инструментов и приборов / В.М. Дмитриев, Т.В. Ганджа, В.В. Ганджа, А.С. Панов // Научная визуализация. – 2016. – Т. 8, № 3. – С. 111–131.

21. Дмитриев В.М. Система виртуальных инструментов и приборов для автоматизации учебных и научных экспериментов / В.М. Дмитриев, Т.В. Ганджа, С.А. Панов // Программные продукты и системы / Software & System. – 2016. – Т. 29, № 3. – С. 154–162.

Дмитриев Вячеслав Михайлович

Профессор кафедры компьютерных систем
в управлении и проектировании (КСУП) ТУСУРа
Ленина пр-т, д. 40, г. Томск, Россия, 634050
Тел.: +7-913-867-01-56
Эл. почта: dmitriewvm@gmail.com

Ганджа Тарас Викторович

Доцент каф. КСУП ТУСУРа
Ленина пр-т, д. 40, г. Томск, Россия, 634050
ORCID: 0000-0002-4996-8114
Тел.: +7-913-846-11-77
Эл. почта: gandgatv@gmail.com

Букреев Александр Сергеевич

Аспирант каф. КСУП ТУСУРа
Ленина пр-т, д. 40, г. Томск, Россия, 634050
Тел.: +7-960-978-86-68
Эл. почта: alexander.eleventh@gmail.com

Dmitriev V.M., Gandzha T.V., Bukreev A.S.

Modeling scenarios to control dynamic objects based on the graphical language X-Robot

As the number of function of dynamic object increases, the algorithm that controls devices becomes more complex. Also, programs written in low-level programming languages of controllers are increasing, that complicates the process of their debugging. The urgency of the development of new means for the formation and modeling of controller operation scenarios is due to the need to take into account many observable variables and the use of a number of control action, but also determined by the non trivial character of the algorithms to operate controllers.

The article discusses the principles of modeling scenarios to control dynamic objects with the use of graphical notations of X-Robot language. In addition to the traditional text representation of a program called a script, a graphic interpretation has been proposed and developed for this language, setting a certain component for its commands. Graphical scenario formation is carried out in the environment of computer simulation MARS at the logical level of a multi-level computer model, at the object level of which the models of a controlled dynamic object are located with the models of executive and measuring devices included in it. This opens up the possibility of forming control scenarios in a graphical form and their preliminary debugging on the model of a dynamic object. At the visual level of the multi-level computer model, control components are located, through which the user has the opportunity to influence the model of the object and the scenario model, as well as the visualization components that display the visualization data for the user, which can be either the values of the observed object variables, or and their generalized parameter-functionals. At present, the developed graphical language for modeling scenarios is adapted to the X-Mega controller, but studies are underway to develop it and apply it to other types of controllers.

Keywords: component, controller, scenario, dynamic object, multilevel computer model

doi: 10.21293/1818-0442-2018-21-2-75-82

References

1. Gumerov R.I. *Praktikum po microprocessoram. Chast' pervaja: microcontrollery AVR. Rukovodstvo*. [Workshop on microprocessors. Part one: microcontrollers AVR. Management]. Kazan': KGU, 2009. 37 p.

2. Evstifeev A.V. *Microcontrollery AVR semeystva Tiny I Mega firmy ATMEL* [Microcontrollers AVR families Tiny and Mega firm ATMEL]. 3-e izd. M.: Izdatel'skiy dom «Dodeka-XXI», 2006. 560 p.

3. Petin V.A. *Proecy s ispol'zovaniem kontrollera Arduino* [Projects using the Arduino controller]. BHV-Peterburg, 2014. 400 p.

4. Demenkov N.P. *Jazyki programirovaniya promyshlennyh controllerov: Uchebnoe posobie* [Programming languages for industrial controllers: Tutorial]. Pod red. K.A. Pupkova. M.: Izd-vo MGTU im. N.E. Bauman publ., 2004. 172 p.

5. IEC 61131-3:2013 | IEC Webstore | water automation, water management, smart city. Available at: <https://webstore.iec.ch/publication/4552> (accessed: 08 May 2018).

6. Marahovskiy V.B., Rozenblyum, A.V. Yakovlev. *Modelirovanie parallel'nyh processov. Seti Petri. Kurs dlja sistemnyh arhitektorov, programmistov, sistemnyh analitikov, proektirovshchikov slozhnyh sistem* [Simulation of parallel processes. Petri net. Course for system architects, programmers, system analysts designers of complex control systems]. SPb.: Professional'naja literatura, IT-Podgotovka publ., 2014. 400 p. (In Russ.).

7. Petrov I.V. *Programmiruemye kontrollery. Standartnye jazyki i prioemy prikladnogo programirovaniya* [Programmable controllers. Standart languages and application programming techniques]. Pod. red. V.P. Djakonova. M.: SOLON-Press, 2004. 256 p. (In Russ.).

8. Home – CODESYS. Available at <https://www.codesys.com/> (accessed: 03 May 2018).

9. ISaGRAF – leading IEC 61131 and IEC 61499 software. Available at <http://www.isagraf.com/index.htm> (accesses: 03.05.2018).

10. Maltsev Yu.I. *Yasyk upravleniya mehanizmami X-Robot* [Language to control of mechanisms X-Robot] // *Elektronnye sredstva i systemy upravleniya*. Tomsk, Tomsk state university of control system and radioelectronics. 2013, no. 2, pp. 114–118.

11. Dmitriev V.M., Shutenkov A.V., Zaychenko T.N., Gandzha T.V. *MARS – sreda modelirovaniya technicheskikh ustroystv i system* [MARS – modeling environment of technical devices and systems]. Tomsk, V-Spectr, 2011. 278 p. (In Russ.).

12. Dmitriev V.M., Gandzha T.V., Bukreev A.S. *Sreda modelirovaniya scenarijev dlja prototipirovaniya controllerov* [Scenario modeling environment for prototyping controllers]. Nauchnaja sessija TUSUR-2017. Materialy Mejdunarodnoy nauchno-technicheskoy konferenzii studentov, aspirantov i molodiy uchenih, posvjaschemmoy 55-letiju TUSURa [Scientific session of TUSUR-2017: materials of the International scientific and technical conference of students, graduate students and young scientists dedicated to the 55th anniversary of TUSUR]. Tomsk. 10–12 May 2017: in 8 parts. Tomsk.: V-Spectr publ., 2017, part 4, pp. 107–110.

13. Dmitriev V.M., Gandzha T.V. *Sreda mnogourovnevnogo komputernogo modelirovaniya himiko-tehnologicheskij system* [Multilevel computer modeling of chemical-technological systems]. Tomsk, Izd-vo Tom. un-ta, 2017, 332 pp.

14. Dmitriev V.M., Gandzha T.V., Gandzha V.V., Maltsev Yu.I. *SVIP – sistema virtual'nyh instrumentov i priborov* [SVIP – system of virtual instruments and devices]. Tomsk, V-Spectr, 2014, 216 p.

15. Dmitriev V.M., Gandzha T.V., Korotina T.Y. *System for visualization and control of computational experi-*

ments in the environment of multilevel modeling. *Reports of TUSUR*, 2010, no. 1 (21), part. 2, pp. 149–155.

16. Dmitriev V.M. Gandzha T.V. Principle of formation of multilevel computer models of SCADA-systems for the control of complex technological objects. *Informatics and control systems*, 2013, no. 2 (36), pp. 24–35.

17. Dmitriev V.M. Shutenkov A.V. Arays L.A. *Avtomatizatsiya modelirovaniya promyshlennyh robotov* [Automation of simulation of industrial robots]. M.: Mashinostroenie, 1995. 304 p. (In Russ.).

18. Dmitriev V.M., Shutenkov A.V., Gandzha T.V. The architecture of a universal computing kernel for the implementation of virtual laboratories. *Instruments and systems. Management. Control, Diagnostic*, 2004, no. 2, pp. 24–28.

19. Grigorieva T.E. Discrete-event modeling in the simulate environment MARS. *Proceedings of TUSUR University*, 2014, no. 1 (31), pp. 152–155.

20. Dmitriev V.M., Gandzha T.V., Gamdza V.V., Panov S.A. Computer simulate of the visual interface in virtual instruments and devices. *Scientific Visualization*, 2016, vol. 8, no. 3, pp. 111–131.

21. Dmitriev V.M., Gandzha T.V., S.A. Panov. System of virtual instruments and devices for automation of education and scientific experiments. *Software & System*, 2016, vol. 29, no. 3, pp. 154–162.

Vjacheslav M. Dmitriev

Doctor of Engineering Sciences, professor,
Department of Computer Control and Design Systems,
Tomsk State University of Control Systems
and Radioelectronics
40, Lenina pr., Tomsk, Russia, 634050
Phone: +7-913-867-01-56
Email: dmitriewm@gmail.com

Taras V. Gandzha

Doctor of Engineering Sciences, Assistant Professor,
Department of Computer Control and Design Systems,
Tomsk State University of Control Systems
and Radioelectronics
40, Lenina pr., Tomsk, Russia, 634050
ORCID 0000-0003-4996-8114
Phone: +7-913-846-11-77
Email: gandgatv@gmail.com

Alexandr A. Bukreev

PhD student Department of Computer Control and Design
Systems, Tomsk State University of Control Systems
and Radioelectronics
40, Lenina pr., Tomsk, Russia, 634050
Phone: +7-960-978 86-68
Email: alexander.eleventh@gmail.com