

Санкт-Петербургский государственный университет информационных технологий,
механики и оптики

Кафедра “Компьютерные технологии”

В.А. Алексеев, А.В. Ларионов

**Сравнение программ управления кофеваркой
”Mark 4 Special Coffee Maker”,
реализованных на основе
нотации Буча и SWITCH-технологии**

Проект создан в рамках
«Движения за открытую проектную документацию»
<http://is.ifmo.ru/>

Содержание

| | |
|--|----|
| 1. Введение..... | 3 |
| 2. Описание кофеварки “Mark 4 Special Coffee Maker”..... | 4 |
| 2.1. Словесное описание..... | 4 |
| 2.2. Описание функций из заголовочного файла “runtime.h”..... | 5 |
| 3. Эмуляция кофеварки..... | 6 |
| 3.1. Введение..... | 6 |
| 3.2. Схема взаимодействия приложений..... | 6 |
| 3.3. Исходный код программы “VisualPart”..... | 6 |
| 4. Программа управления кофеваркой “Mark 4 Special Coffee Maker” из книги Р. Мартина..... | 7 |
| 4.1. Описание программы управления..... | 7 |
| 4.1.1. Постановка задачи..... | 7 |
| 4.1.2. Описание структуры программы..... | 8 |
| 4.1.3. Класс “MarkIV_Warmer”..... | 10 |
| 4.1.4. Класс “MarkIV_Sprayer”..... | 11 |
| 4.1.5. Класс “MarkIV_UI”..... | 11 |
| 4.2. Исходный код..... | 12 |
| 5. Программы управления кофеваркой “Mark 4 Special Coffee Maker”, написанные с использованием SWITCH-технологии..... | 13 |
| 5.1. Программа управления кофеваркой с одним автоматом..... | 13 |
| 5.1.1. Структурная схема программы управления кофеваркой с одним автоматом..... | 13 |
| 5.1.2. Автомат А0..... | 14 |
| 5.1.2.1. Перечень событий..... | 14 |
| 5.1.2.2. Перечень выходных воздействий..... | 14 |
| 5.1.2.3. Схема связей..... | 14 |
| 5.1.2.4. Граф переходов..... | 15 |
| 5.1.3. Исходный код..... | 16 |
| 5.2. Программа управления кофеваркой с тремя автоматами..... | 17 |
| 5.2.1. Структурная схема программы управления кофеваркой с тремя автоматами..... | 17 |
| 5.2.2. Автомат А1..... | 18 |
| 5.2.2.1. Перечень событий..... | 18 |
| 5.2.2.2. Перечень выходных воздействий..... | 18 |
| 5.2.2.3. Схема связей..... | 18 |
| 5.2.2.4. Граф переходов..... | 19 |
| 5.2.3. Автомат А2..... | 20 |
| 5.2.3.1. Перечень событий..... | 20 |
| 5.2.3.2. Перечень выходных воздействий..... | 20 |
| 5.2.3.3. Схема связей..... | 20 |
| 5.2.3.4. Граф переходов..... | 20 |
| 5.2.4. Автомат А3..... | 21 |
| 5.2.4.1. Перечень событий..... | 21 |
| 5.2.4.2. Перечень входных переменных..... | 21 |
| 5.2.4.3. Перечень выходных воздействий..... | 21 |
| 5.2.4.4. Схема связей..... | 21 |
| 5.2.4.5. Граф переходов..... | 21 |
| 5.2.5. Исходный код..... | 22 |
| 6. Заключение..... | 23 |
| Литература..... | 23 |

1. Введение

При проектировании программ управления техническими устройствами используются различные технологии и формы записи алгоритмов управления. В данной работе для сравнения подходов к проектированию рассмотрен пример программы управления кофеваркой “Mark 4 Special Coffee Maker” из работы [1], и написаны две программы управления этой кофеваркой, которые спроектированы, используя SWITCH-технологию [2].

В книге Р. Мартина [1] пример программы управления кофеваркой описан, используя нотацию Г. Буча. Однако этот пример не содержит проектной документации. Рисунки и диаграммы не доведены до вида математических моделей и не описывают точное поведение программы. В виду того, что полностью воссоздать процесс написания программы и проектную документацию в этом случае невозможно, то эта программа, исходный код которой взят из книги (его также можно скачать по адресу <http://flinflon.brandonu.ca/dueck/2000/62286/martin.zip>), лишь описана в данной работе.

Из программ, созданных по SWITCH-технологии, в первой используется один автомат, а во второй (также как и в программе из книги Мартина) – три автомата. Автоматы для этих программ были описаны в Microsoft® Visio®, а исходный код был автоматически сгенерирован программой “Visio2SWITCH” (ее можно скачать с сайтов <http://www.geocities.com/goloveshin/> и <http://is.ifmo.ru/>). Благодарим ее автора, Александра Головешина.

2. Описание кофеварки “Mark 4 Special Coffee Maker”

2.1. Словесное описание

Основные компоненты кофеварки “Mark 4 Special Coffee Maker”:

- резервуар, в котором кипятится вода. Он в дальнейшем называется “кипятильник”. Кипятильник может быть включен или выключен;
- сенсор, определяющий уровень воды в кипятильнике;
- клапан сброса давления – может быть открыт или закрыт;
- чашка под кофе (далее чашка) – может вместить до 12 обычных чашек кофе;
- подогревательная подставка для чашки – может быть включена или выключена;
- сенсор, определяющий есть ли чашка на подогревательной подставке. Если да, то он также определяет, наполнена ли чашка жидкостью;
- кнопка начала цикла варки кофе. При нажатии она запоминает свое состояние до первого ее опроса;
- индикатор окончания цикла варки кофе.

Цикл варки кофе состоит в следующем:

- до начала цикла варки кофе кипятильник выключен, клапан сброса давления закрыт, подогревательная подставка выключена, индикатор окончания цикла варки кофе выключен;
- пользователь наполняет кипятильник водой и ставит пустую чашку на подогревательную подставку. После этого он нажимает кнопку начала цикла варки кофе;
- если при нажатии кнопки кипятильник пуст или на подогревательной подставке нет пустой чашки, то цикл варки кофе не начинается. В противном случае кипятильник включается, и в нем начинает кипятиться вода;
- когда вода вскипела, она под давлением пара выливается из кипятильника, проходит через фильтр с молотым кофе и попадает в чашку;
- как только кофе начинает течь в чашку, сенсор подогревательной подставки определяет, что в чашке есть кофе, и включается подогревательная подставка;
- когда в кипятильнике заканчивается вода, его сенсор определяет, что вода закончилась, кипятильник выключается, и загорается индикатор окончания цикла варки кофе;
- индикатор окончания цикла варки кофе выключается, как только на подставку ставят пустую чашку. В этот момент цикл варки кофе заканчивается;
- если во время цикла варки кофе пользователь убирает подогреваемую чашку с подогревательной подставки, то подставка выключается. Если в этот момент вода кипятится, то открывается клапан сброса давления, и выключается кипятильник. Если во время цикла варки кофе пользователь ставит пустую чашку на подогревательную подставку, то подставка включается. Если в этот момент в кипятильнике есть вода, и индикатор окончания цикла варки кофе выключен, то закрывается клапан сброса давления, и включается кипятильник.

Управление кофеваркой осуществляется с помощью программы управления, загружаемой в память контроллера кофеварки. Программа управления использует заголовочный файл “runtime.h” на языке “C”, который содержит объявления функций для управления кофеваркой. В бесконечном цикле программы управления вызываются эти функции, происходит опрос состояния сенсоров кофеварки и изменение состояния компонентов кофеварки.

2.2. Описание функций из заголовочного файла “runtime.h”

| | |
|-------------|---|
| Имя: | GetWarmerStatus |
| Объявление: | enum WarmerPlateStatus { potNotEmpty, popEmpty, warmerEmpty }; WarmerPlateStatus GetWarmerPlateStatus(); |
| Описание: | Возвращает состояние сенсора подогревательной подставки. Сенсор подогревательной подставки определяет находится ли на ней чашка, и если находится, то, есть ли в чашке вода или нет. |
| Имя: | GetBoilerStatus |
| Объявление: | enum BoilerStatus { boilerEmpty, boilerNotEmpty }; BoilerStatus GetBoilerStatus(); |
| Описание: | Возвращает состояние сенсора кипятильника. boilerEmpty, если в кипятильнике воды меньше, чем половина обычной чашки, boilerNotEmpty – если больше. |
| Имя: | GetBrewButtonStatus |
| Объявление: | enum BrewBoilerStatus { brewButtonPushed, brewButtonNotPushed }; BrewButtonStatus GetBrewButtonStatus(); |
| Описание: | Возвращает состояние кнопки начала цикла варки кофе. После нажатия пользователем на кнопку, первый вызов этой функции вернет значение brewButtonPushed (кнопка нажата). После этого состояние кнопки сбросится. Следующие вызовы этой функции будут возвращать значение brewButtonNotPushed до тех пор, пока кнопка снова не будет нажата. |
| Имя: | SetBoilerState |
| Объявление: | enum BoilerHeaterState { boilerOn, boilerOff }; void SetBoilerState(BoilerHeaterState); |
| Описание: | Функция включает (при входном воздействии boilerOn) или выключает (при входном воздействии boilerOff) кипятильник. |
| Имя: | SetWarmerState |
| Объявление: | enum WarmerState { warmerOn, warmerOff }; void SetWarmerState(WarmerState); |
| Описание: | Функция включает (warmerOn) или выключает (warmerOff) подогревательную подставку. |
| Имя: | SetIndicatorState |
| Объявление: | enum IndicatorState { indicatorOn, indicatorOff }; void SetIndicatorState(IndicatorState); |
| Описание: | Функция включает (indicatorOn) или выключает (indicatorOff) индикатор окончания цикла варки кофе. |
| Имя: | SetReliefValveState |
| Объявление: | enum ReliefValveState { valveOpen, valveClosed }; void SetReliefValveState(ReliefValveState); |
| Описание: | Функция открывает (valveOpen) или закрывает (valveClosed) клапан сброса давления. Когда этот клапан закрыт, давление пара в кипятильнике заставит горячую воду течь через фильтр в чашку. Когда клапан открыт, давление в кипятильнике падает, и вода из кипятильника не выливается. |

3. Эмуляция кофеварки

3.1. Введение

Для эмуляции кофеварки был использован следующий подход. Программная и аппаратная части были разнесены по разным исполняемым модулям. Связь между ними устанавливается с помощью технологии COM, при использовании которой Windows-приложения разделяются на серверы и клиенты. Приложение “СОМ-сервер” содержит в себе СОМ-объекты и их описания (методы, свойства и т.д.). Приложение “СОМ-клиент” создает объекты “СОМ-сервера” и использует их, зная их методы и свойства.

Программа “VisualPart” (приложение “СОМ-сервер”) визуализирует работу кофеварки. Она создана в среде разработки “Visual Basic 6.0”. Визуализатор кофеварки является СОМ-объектом, описанным в “VisualPart”.

Программа “WorkingPart” (приложение “СОМ-клиент”) создает объект-визуализатор кофеварки из исполняемого модуля “VisualPart”, а затем управляет им. Эта программа создана в среде разработки “Visual C++ 6.0”. Она с незначительными изменениями (удаление кода, связанного с созданием объекта-визуализатора) может быть использована для управления кофеваркой “Mark 4 Special Coffee Maker”.

3.2. Схема взаимодействия приложений

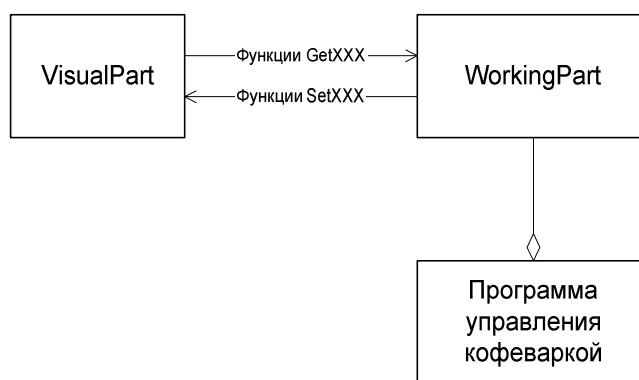


Рис. 1. Схема взаимодействия программ

3.3. Исходный код программы “VisualPart”

Исходный код программы можно найти в архиве “sources.zip” в каталоге “VisualPart.vb”. Он содержит следующие файлы:

- “Mark4.vbp” - файл проекта “Visual Basic 6.0”;
- “Mark4Visualisation.cls” - файл класса “Mark4CoffeeMaker”;
- “VForm.frm” - файл формы, визуализирующей кофеварку.

4. Программа управления кофеваркой “Mark 4 Special Coffee Maker” из книги Р. Мартина

4.1. Описание программы управления

4.1.1. Постановка задачи

Программа управления кофеваркой “Mark 4 Special Coffee Maker” должна соответствовать диаграмме переходов (“State Transition Diagram” или STD) в нотации Д. Харела (рис. 2).

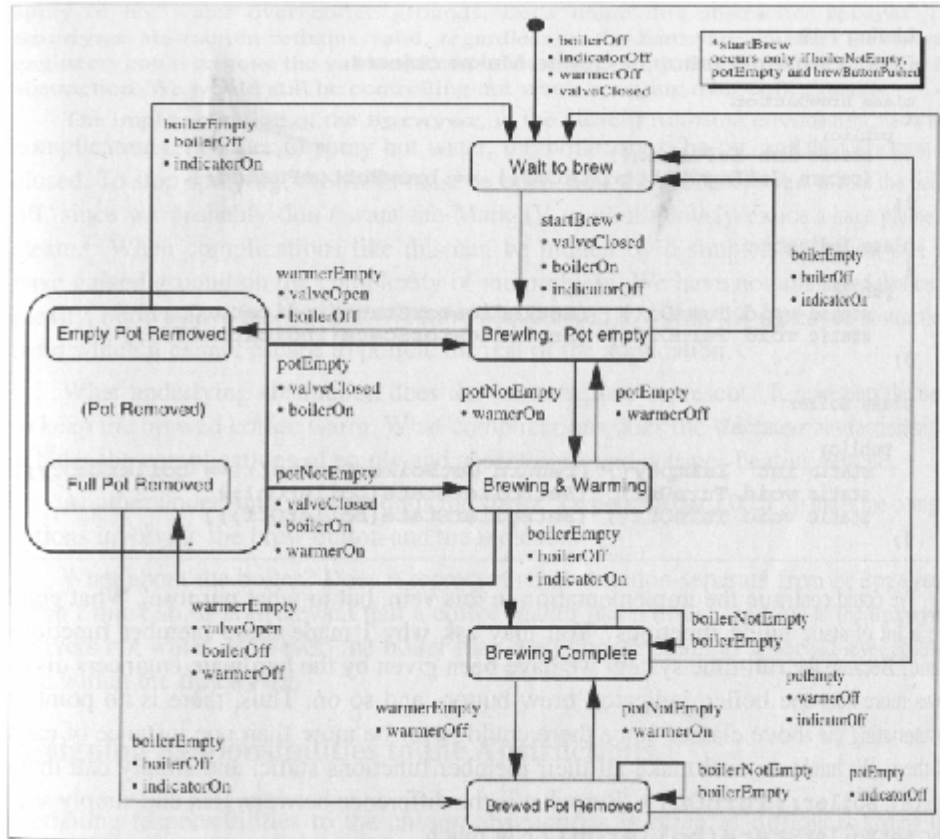


Рис. 2. Диаграмма переходов для программы управления кофеваркой “Mark 4 Special Coffee Maker” в нотации Харела

На рис. 2 прямоугольники обозначают состояния автомата, а стрелки – переходы из состояния в состояние. Названия рядом со стрелками, не помеченные точками, обозначают события (например, “boilerEmpty” – произошло событие “кипятильник пуст”). Каждое из событий соответствует изменению состояния одного из сенсоров кофеварки, за исключением события “startBrew” (помечено звездочкой). Оно происходит тогда и только тогда, когда выполняются три условия: кипятильник не пуст (“boilerNotEmpty”), пустая чашка находится на нагревательной подставке (“potEmpty”) и нажата кнопка запуска (“brewButtonPushed”). Названия рядом со стрелками, помеченные точками, означают вызов методов для установки состояния элементов кофеварки (например, метод “warmerOff” обеспечивает выключение подогревательной подставки).

4.1.2. Описание структуры программы

Хотя можно было бы написать программу управления непосредственно по данной диаграмме переходов, в работе [1] предлагается разделить логику работы программы между объектами, соответствующими аппаратным частям кофеварки. В качестве таких объектов были выбраны: “Sprayer” (управляет кипятильником и заполнением чашки), “Warmer” (управляет подогревательной подставкой) и “UI” (реализует интерфейс пользователя). Предполагается, что такое разделение в дальнейшем может помочь при модификации программы.

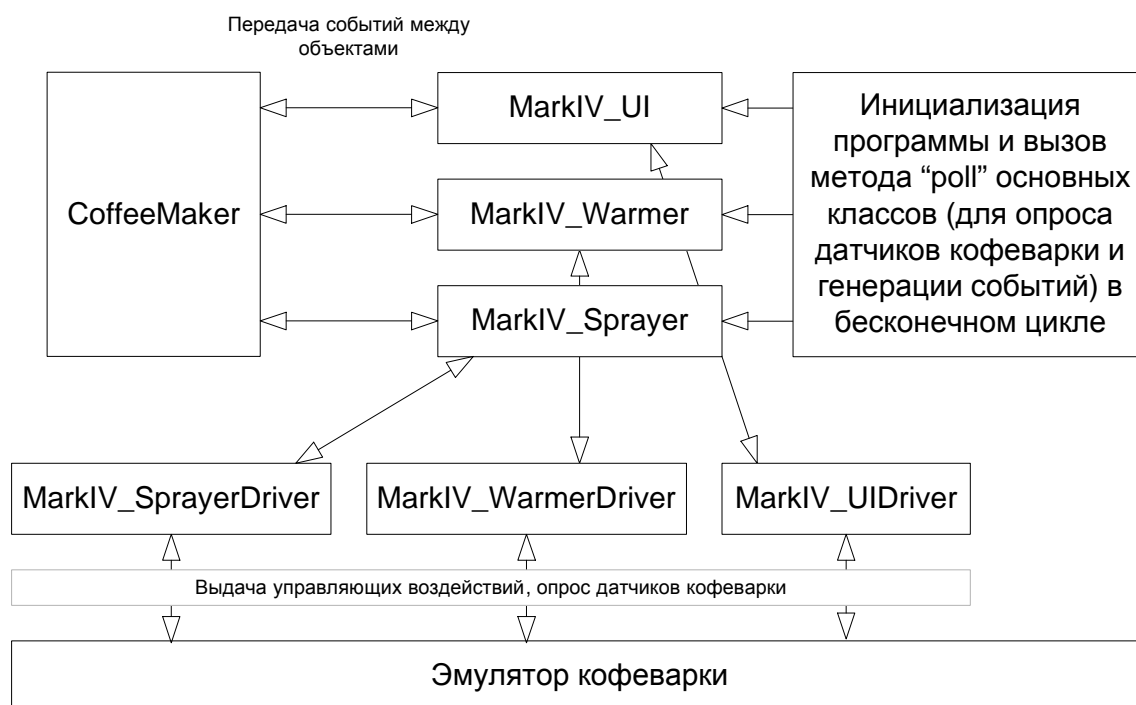


Рис. 3. Структурная схема программы управления кофеваркой для примера из книги Р. Мартина

В программе логика работы разделена между классами “MarkIV_Warmer”, “MarkIV_Sprayer” и “MarkIV_UI”. В дальнейшем будем называть их основными классами, которые напоминают автоматные классы, если использовать терминологию SWITCH-технологии.

Каждый из основных классов включает в себя ссылки на класс-драйвер и класс-клиент. Для класса “MarkIV_Warmer” – это “MarkIV_WarmerDriver” и “CoffeeMaker”, для класса “MarkIV_Sprayer” – “MarkIV_SprayerDriver” и “CoffeeMaker”, а для класса “MarkIV_UI” – “MarkIV_UIDriver” и “CoffeeMaker”.

Взаимодействие основных классов осуществляется с помощью класса-надстройки “CoffeeMaker”, который содержит указатели на каждый из этих классов. Он также является клиентом для каждого из основных классов и наследован от абстрактных классов “WarmerClient”, “SprayerClient” и “UIClient”.

Каждый из методов класса-клиента соответствует событию от одного из основных классов. При вызове такого метода класс “CoffeeMaker” вызовет методы других основных классов, соответствующие этому событию.

Методы классов-драйверов непосредственно взаимодействуют с аппаратной частью кофеварки, например, устанавливают состояние элементов или опрашивают сенсоры.

На рис. 4 изображена диаграмма классов для программы в нотации Г. Буча.

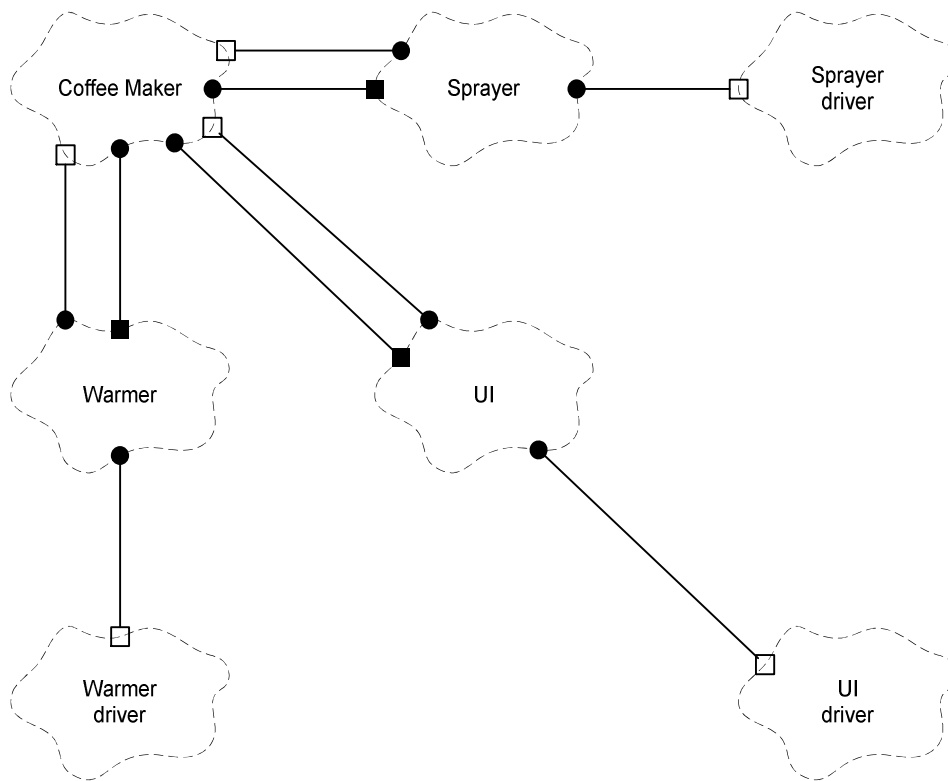


Рис. 4. Диаграмма классов в нотации Г. Буча для программы управления кофеваркой “Mark 4 Special Coffee Maker”

Программа начинает работу с создания объекта “CoffeeMaker”, затем объектов драйверов “MarkIV_WarmerDriver”, “MarkIV_SprayerDriver” и “MarkIV_UIDriver”, а затем “MarkIV_Warmer”, “MarkIV_Sprayer” и “MarkIV_UI”. Конструктору каждого из этих классов передаются соответствующий объект-драйвер и объект “CoffeeMaker” в качестве объекта-клиента.

Далее объекту “CoffeeMaker” передаются “MarkIV_Warmer”, “MarkIV_Sprayer” и “MarkIV_UI”. После чего в бесконечном цикле у объектов “MarkIV_Warmer”, “MarkIV_Sprayer” и “MarkIV_UI” вызывается метод “Poll” (опрос). В этом “Poll” основные объекты опрашивают свои объекты-драйверы о состоянии кофеварки и в зависимости от него и от своего собственного состояния генерируют события.

В следующих разделах следует описание основных классов. На диаграммах переходов для классов название рядом со стрелкой без точки означает название события, после которого происходит переход. Если название события начинается с маленькой буквы, то это событие вызывается методами данного класса, если с большой буквы, то методами класса-надстройки “CoffeeMaker”. Название рядом со стрелкой с точкой, начинающееся с маленькой буквы, означает вызов одноименного метода класса-драйвера; а начинающееся с большой буквы - вызов одноименного метода класса-клиента.

4.1.3. Класс “MarkIV_Warmer”

Класс “MarkIV_Warmer” управляет подогревательной подставкой под чашкой. На рис. 5 изображена диаграмма переходов для этого класса в нотации Харела.

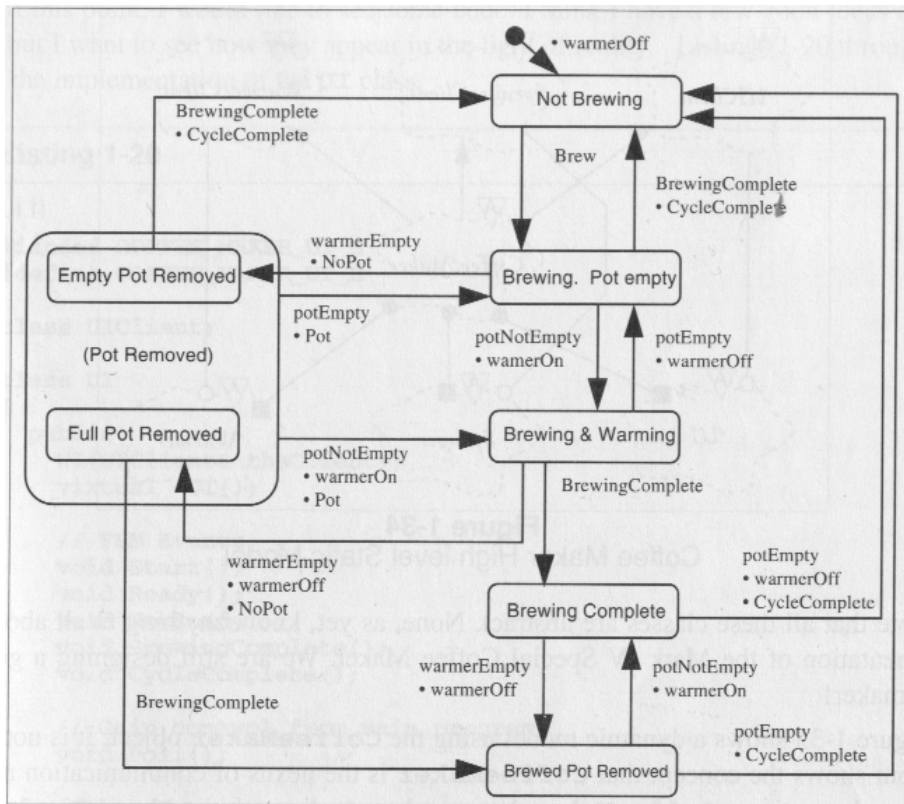


Рис. 5. Диаграмма переходов для класса “MarkIV_Warmer” в нотации Харела

4.1.4. Класс “MarkIV_Sprayer”

Класс “MarkIV_Sprayer” управляет процессом наливания кофе в чашку. На рис. 6 изображена диаграмма переходов для этого класса в нотации Харела.

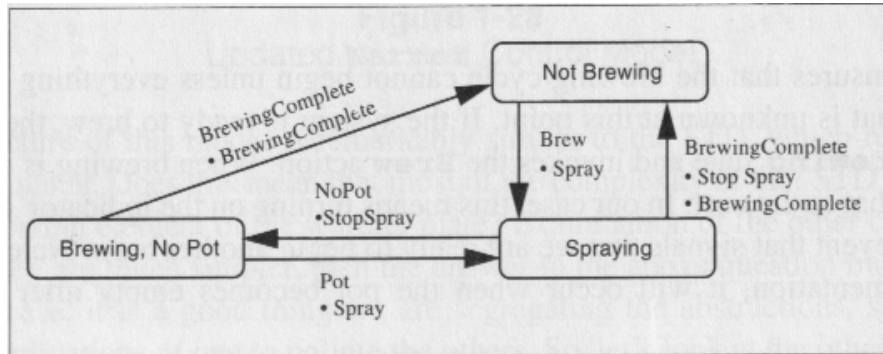


Рис. 6. Диаграмма переходов для класса “MarkIV_Sprayer” в нотации Харела

4.1.5. Класс “MarkIV_UI”

Класс “MarkIV_UI” управляет запуском цикла приготовления кофе. На рис. 7 изображена диаграмма переходов для этого класса в нотации Харела.

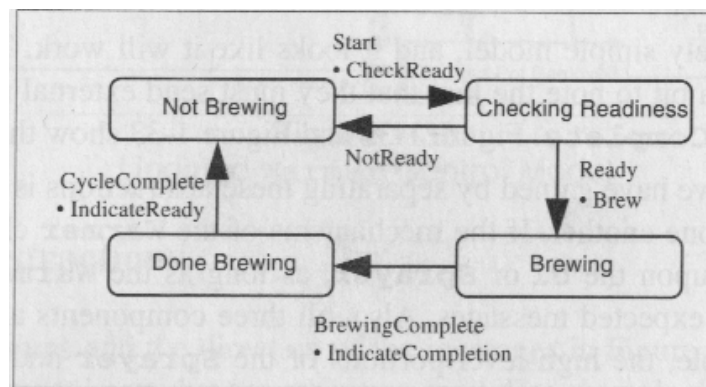


Рис. 7. Диаграмма переходов для класса “MarkIV_UI” в нотации Харела

4.2. Исходный код

Исходный код программы можно найти в архиве “sources.zip” в каталоге “WorkingPart”. Он содержит следующие файлы:

- “WorkingPart.dsw” и “WorkingPart.dsp” - файлы проекта “Visual C++ 6.0”;
- “WorkingPart.cpp” - файл содержащий точку входа программы;
- “sprayer.h” и “sprayer.cpp” - файлы абстрактного интерфейса объекта “Sprayer”;
- “warmer.h” и “warmer.cpp” - файлы абстрактного интерфейса объекта “Warmer”;
- “ui.h” и “ui.cpp” - файлы абстрактного интерфейса объекта “UI”;
- “m4Sprayer.h” и “m4Sprayer.cpp” - файлы реализации объекта “Sprayer”;
- “m4Warmer.h” и “m4Warmer.cpp” - файлы реализации объекта “Warmer”;
- “markIV_UI.h” и “markIV_UI.cpp” - файлы реализации объекта “UI”;
- “sprayerDriver.h” - файл абстрактного интерфейса класса-драйвера для объекта “Sprayer”;
- “warmerDriver.h” - файл абстрактного интерфейса класса-драйвера для объекта “Warmer”;
- “uiDriver.h” - файл абстрактного интерфейса класса-драйвера для объекта “UI”;
- “sprayerClient.h” - файл абстрактного интерфейса класса-клиента для объекта “Sprayer”;
- “warmerClient.h” - файл абстрактного интерфейса класса-клиента для объекта “Warmer”;
- “uiClient.h” - файл абстрактного интерфейса класса-клиента для объекта “UI”;
- “m4SpDriver.h” и “m4SpDriver.cpp” - файлы реализации класса-драйвера для объекта “Sprayer”;
- “m4WmDriver.h” и “m4WmDriver.cpp” - файлы реализации класса-драйвера для объекта “Warmer”;
- “m4UIDriver.h” и “m4UIDriver.cpp” - файлы реализации класса-драйвера для объекта “UI”;
- “coffeemaker.h” и “coffeemaker.cpp” - файлы реализации класса-надстройки “CoffeeMaker”;
- “runtime.h” - заголовочный файл содержащий объявления функций для управления кофеваркой.

5. Программы управления кофеваркой “Mark 4 Special Coffee Maker”, написанные с использованием SWITCH-технологии

5.1. Программа управления кофеваркой с одним автоматом

5.1.1. Структурная схема программы управления кофеваркой с одним автоматом

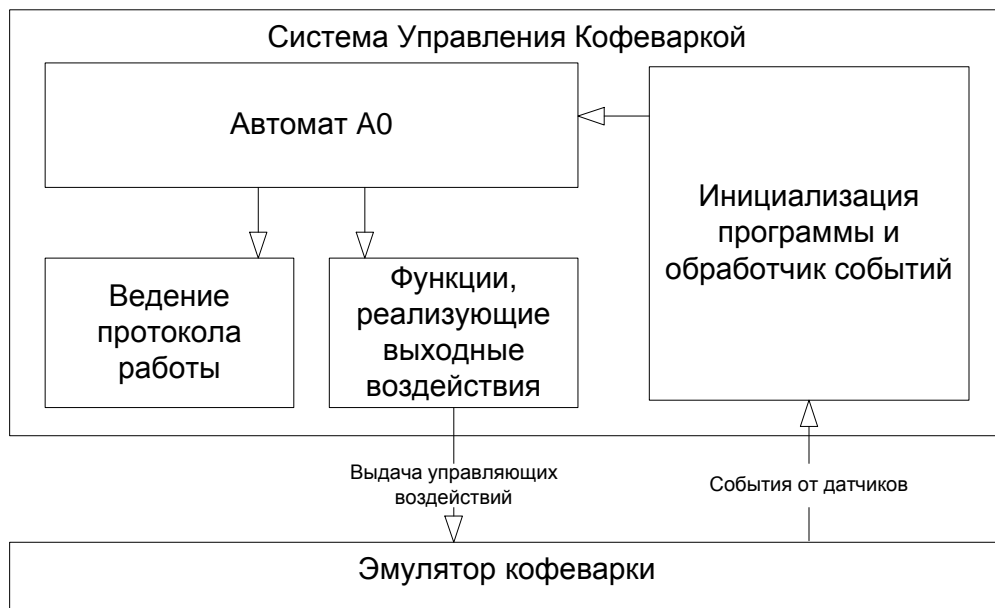


Рис. 8. Структурная схема программы управления кофеваркой с одним автоматом

5.1.2. Автомат А0

Схема связей автомата А0 приведена на рис. 9, а его граф переходов, построенный по диаграмме в нотации Харела, представленной на рис. 2 – на рис. 10.

Начальное состояние автомата нулевое.

5.1.2.1. Перечень событий

| | |
|----|---|
| 10 | Начать приготовление кофе |
| 20 | Кипятильник пустой |
| 30 | Подогревательная подставка пустая |
| 31 | На подогревательной подставке пустая чашка |
| 32 | На подогревательной подставке не пустая чашка |

5.1.2.2. Перечень выходных воздействий

| | |
|-----|--------------------------------------|
| 100 | Включить кипятильник |
| 101 | Выключить кипятильник |
| 110 | Включить подогревательную подставку |
| 111 | Выключить подогревательную подставку |
| 120 | Открыть клапан сброса давления |
| 121 | Закрыть клапан сброса давления |
| 130 | Включить индикатор окончания работы |
| 131 | Выключить индикатор окончания работы |

5.1.2.3. Схема связей

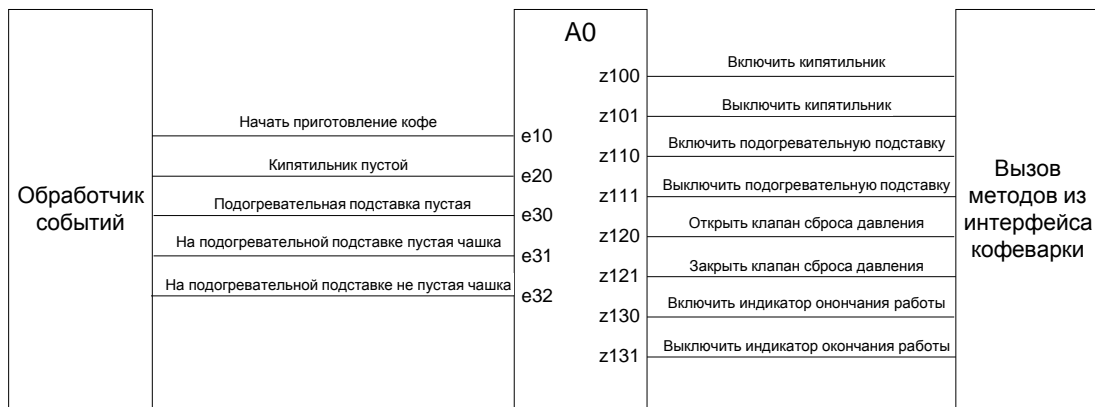


Рис. 9. Схема связей автомата А0

5.1.2.4. Граф переходов

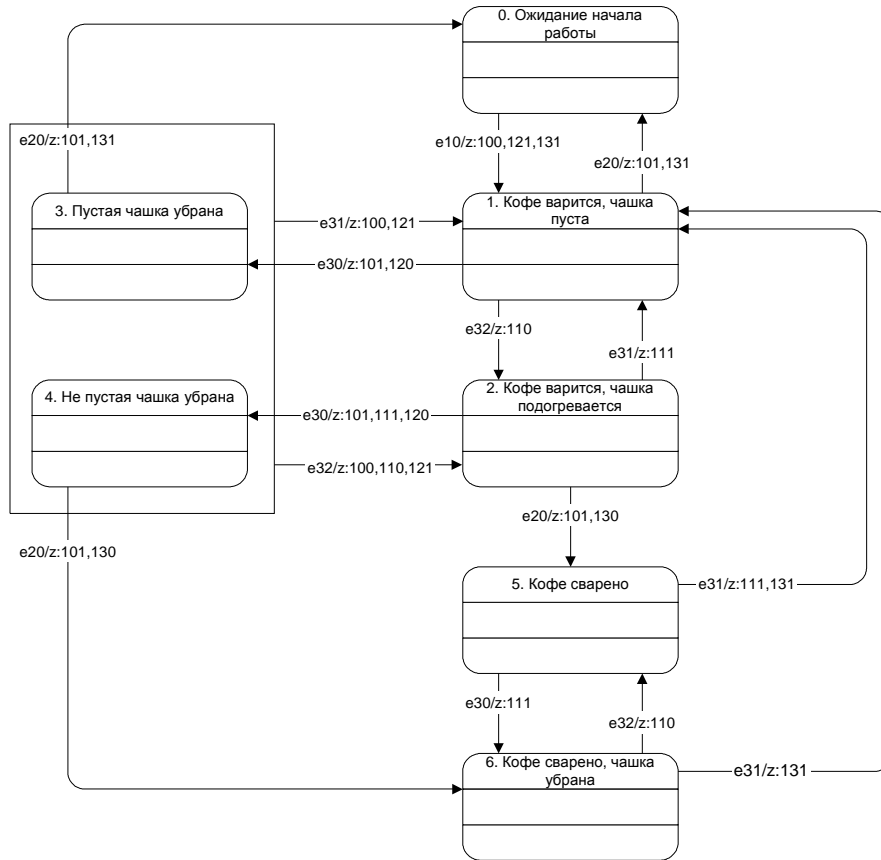


Рис. 10. Граф переходов автомата A0

5.1.3. Исходный код

Исходный код программы можно найти в архиве “sources.zip” в каталоге “1AUTSample”. Он содержит следующие файлы:

- “1AUTSample.dsw” и “1AUTSample.dsp” - файлы проекта “Visual C++ 6.0”;
- “main.cpp” - файл содержащий точку входа программы;
- “runtime.h” - заголовочный файл содержащий объявления функций для управления кофеваркой.
- “types.h” - файл определения типов;
- “common.h” - файл объявления автоматных функций, переменных состояний автоматов;
- “common.cpp” - файл реализаций автоматных функций;
- “x.cpp” - файл реализации функций входных переменных как “обертки” функций пользователя;
- “x_user.cpp” - файл реализации пользовательских функций входных переменных;
- “z.cpp” - файл реализации функций действий как “обертки” функций пользователя;
- “z_user.cpp” - файл реализации пользовательских функций действий;
- “log.h” - файл отладочной информации;
- “log.cpp” - файл реализации отладочных функций как “обертки” функций пользователя;
- “log_user.cpp” - файл реализации пользовательских отладочных функций;

5.2. Программа управления кофеваркой с тремя автоматами

5.2.1. Структурная схема программы управления кофеваркой с тремя автоматами

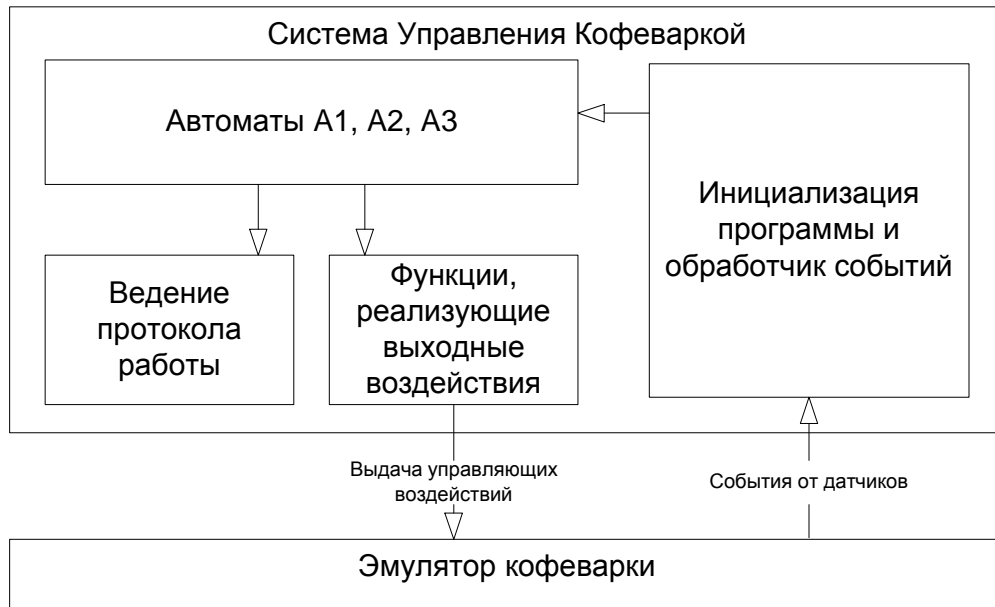


Рис. 11. Структурная схема программы управления кофеваркой с тремя автоматами

Автоматы A1, A2, A3 могут вызывать друг друга, но гарантируется, что если автомат A_i вызвал автомат A_j , то на этом шаге автомат A_j уже не вызовет другой автомат. Хотя конструкция, в которой не существует древовидной иерархии автоматов и не очень удобна, но поскольку она присутствует в примере из книги Мартина, оставим ее без изменений и в этой реализации.

5.2.2. Автомат A1

Схема связей автомата A1 приведена на рис. 12, а его граф переходов, соответствующий классу “MarkIV_Warmer” из примера Мартина – на рис. 13.

Начальное состояние автомата нулевое.

5.2.2.1. Перечень событий

| | |
|----|---|
| 10 | Начать цикл приготовления кофе |
| 20 | Кофе сварено |
| 30 | Подогревательная подставка пустая |
| 31 | На подогревательной подставке пустая чашка |
| 32 | На подогревательной подставке не пустая чашка |

5.2.2.2. Перечень выходных воздействий

| | |
|-----|--------------------------------------|
| 100 | Включить подогревательную подставку |
| 101 | Выключить подогревательную подставку |

5.2.2.3. Схема связей



Рис. 12. Схема связей автомата A1

5.2.2.4. Граф переходов

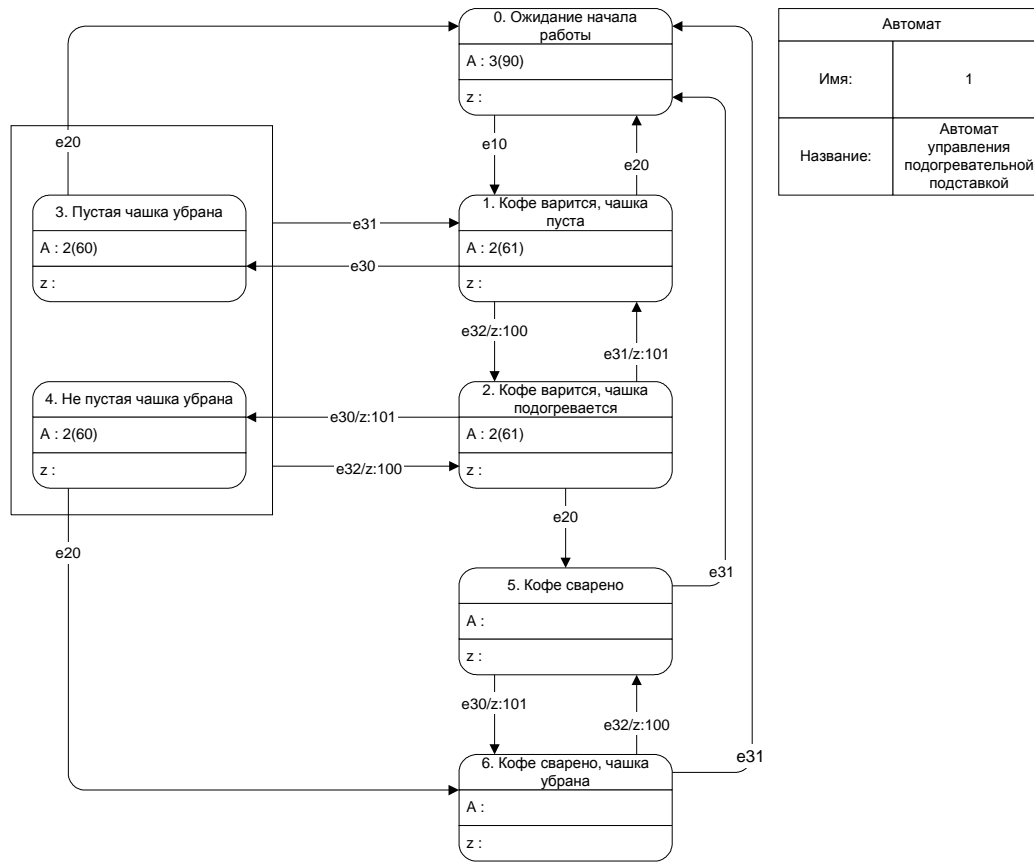


Рис. 13. Граф переходов автомата А1

5.2.3. Автомат A2

Схема связей автомата A2 приведена на рис. 14, а его граф переходов, соответствующий классу “MarkIV_Sprayer” из примера Мартина – на рис. 15.

Начальное состояние автомата нулевое.

5.2.3.1. Перечень событий

- 40 Начать цикл приготовления кофе
- 50 Кофе сварено
- 60 Подогревательная подставка пустая
- 61 Подогревательная подставка не пустая

5.2.3.2. Перечень выходных воздействий

- 110 Прекратить налив кофе в чашку
- 111 Наливать кофе в чашку

5.2.3.3. Схема связей



Рис. 14. Схема связей автомата A2

5.2.3.4. Граф переходов

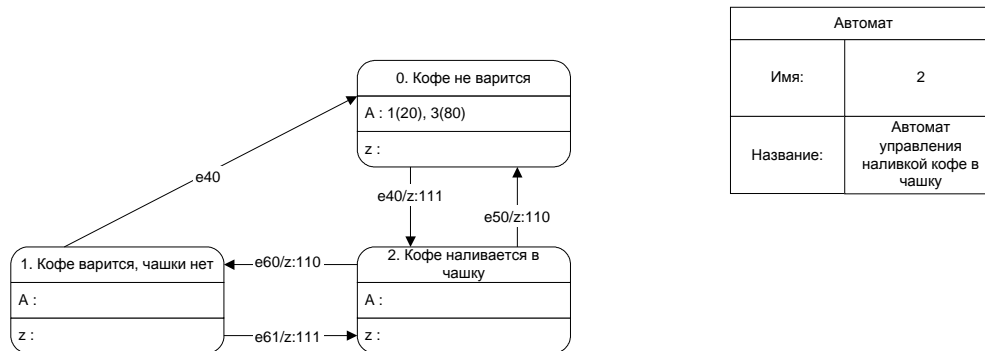


Рис. 15. Граф переходов автомата A2

5.2.4. Автомат А3

Автомат реализует функциональность, соответствующую классу “MarkIV_UI” из примера Мартина.

У автомата (рис. 7) убрано одно состояние “Checking Readiness”, так как в нем автомат находится только при вызове функции *CheckReady*, которая, в свою очередь, заменена входной переменной *x1*, равная единице, когда нажата кнопка начала работы, кипятильник не пустой и на подогревательной подставке находится пустая чашка, и ноль – в противном случае.

Схема связей автомата А3 приведена на рис. 16, а его граф переходов – на рис. 17. Начальное состояние автомата нулевое.

5.2.4.1. Перечень событий

| | |
|----|----------------------------------|
| 70 | Начать цикл приготовления кофе |
| 80 | Кофе сварено |
| 90 | Цикл приготовления кофе завершен |

5.2.4.2. Перечень входных переменных

| | |
|---|----------------------|
| 1 | Готовность к запуску |
|---|----------------------|

5.2.4.3. Перечень выходных воздействий

| | |
|-----|--------------------------------------|
| 120 | Включить индикатор окончания работы |
| 121 | Выключить индикатор окончания работы |

5.2.4.4. Схема связей

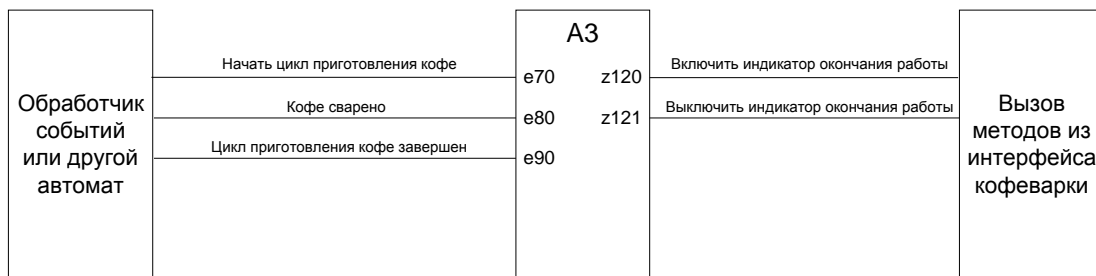


Рис. 16. Схема связей автомата А3

5.2.4.5. Граф переходов

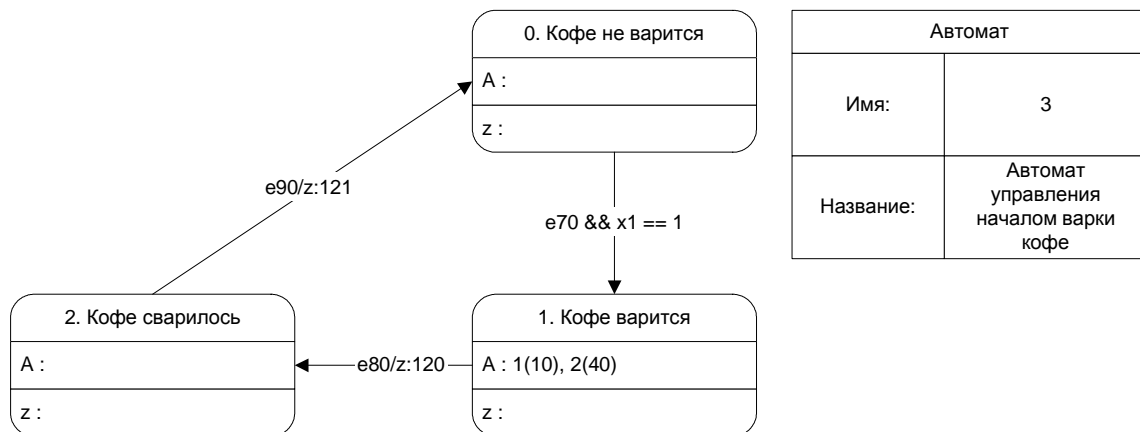


Рис. 17. Граф переходов автомата А3

5.2.5. Исходный код

Исходный код программы можно найти в архиве “sources.zip” в каталоге “3AUTSample”. Он содержит следующие файлы:

- “3AUTSample.dsw” и “3AUTSample.dsp” - файлы проекта “Visual C++ 6.0”;
- “main.cpp” - файл содержащий точку входа программы;
- “runtime.h” - заголовочный файл содержащий объявления функций для управления кофеваркой.
- “types.h” - файл определения типов;
- “common.h” - файл объявления автоматных функций, переменных состояний автоматов;
- “common.cpp” - файл реализаций автоматных функций;
- “x.cpp” - файл реализации функций входных переменных как “оберток” функций пользователя;
- “x_user.cpp” - файл реализации пользовательских функций входных переменных;
- “z.cpp” - файл реализации функций действий как “оберток” функций пользователя;
- “z_user.cpp” - файл реализации пользовательских функций действий;
- “log.h” - файл отладочной информации;
- “log.cpp” - файл реализации отладочных функций как “оберток” функций пользователя;
- “log_user.cpp” - файл реализации пользовательских отладочных функций;

6. Заключение

В заключение сравним рассмотренные подходы.

Документация в примере Мартина присутствует в виде нескольких высокоуровневых диаграмм в нотации Буча. Хотя эти диаграммы дают представление о структуре кода программы, но они мало говорят о ее поведении и взаимодействии объектов. Несмотря на то, что программа из книги Мартина построена по диаграмме переходов, структура программы имеет с ней мало общего. Поэтому эту диаграмму вряд ли стоит считать частью документации программы.

Примеры с использованием SWITCH-технологии построены так, что их код практически полностью соответствует диаграммам переходов, более того – автоматически генерируется программой “visio2switch”.

Может показаться что, так как логика работы программы из книги Мартина разделена между независимыми объектами, то в нее будет удобно вносить изменения. Однако, если при таком проектировании программы техническое задание будет изменено, то придется практически полностью повторить процесс ее проектирования.

Видимо, для такого рода задач SWITCH-технология, которая предлагает алгоритм построения программ и стандартную форму документирования и реализации, предпочтительнее.

Литература

1. *Мартин Р.* Designing Object-Oriented C++ Applications Using The Booch Method. NJ: Prentice Hall, 1993.
2. *Шальто А. А.* SWITCH-технология. Алгоритмизация и программирование задач логического управления. СПб.: Наука, 1998.