

Санкт-Петербургский государственный университет информационных
технологий, механики и оптики

Кафедра «Компьютерные технологии»

Ю. К. Чеботарева

**Визуальное редактирование автоматов в
«Виртуальной лаборатории для первоначального
обучения проектированию программ»**

Проектная документация

Санкт-Петербург

2008

Введение	3
1. Обзор	4
1.1. Режим редактирования	4
1.2. Режим навигации	5
1.3. Сохранение и загрузка графов	5
2. Пример использования	7
2.1. Постановка задачи	7
2.2. Исследование задачи в виртуальной лаборатории	8
Заключение	10
Источники	11

Введение

Использование проектного подхода является одним из современных эффективных методов обучения при подготовке специалистов в области разработки программного обеспечения [1]. При практической реализации идеи проектного подхода возникает задача выбора парадигмы программирования, на базе которой строится учебный процесс.

Кафедры компьютерных технологий и технологий программирования СПбГУ ИТМО активно используют данный подход в работе со студентами, начиная уже с первого курса. Опыт их работы показывает, что приемлемым решением является использование автоматного программирования в качестве основы для обучения проектированию программ. Для поддержки автоматного программирования создано инструментальное средство *UniMod* [2]. Однако для первоначального знакомства с проектированием это средство является достаточно сложным.

В работе [3] предложен упрощенный подход к обучению проектированию программ, основанный на автоматном программировании. Авторами работы [3] разработана *«Виртуальная лаборатория для первоначального обучения проектированию программ»* [5], в которой учащимся предлагается (по спроектированному самостоятельно графу переходов) реализовать поведение различных сущностей на предложенном текстовом языке автоматного программирования.

В рамках настоящей работы рассматривается созданный автором компонент, позволяющий задавать непосредственно граф переходов. Использование этого компонента в *«Виртуальной лаборатории для первоначального обучения проектированию программ»* повысит наглядность и значительно упростит поиск ошибок в построенных учащимися автоматах.

1. Обзор

В качестве альтернативы текстовому языку автоматного программирования автором разработан компонент для непосредственного задания графа переходов. Для разработки указанного компонента использовался язык *Java*. Таким образом, он может без особых усилий быть встроен в «Виртуальную лабораторию для первоначального обучения проектированию программ».

Рассмотрим возможности указанного компонента более подробно. Он представляет собой редактор графов переходов и содержит область редактирования, кнопки сохранения и загрузки графов, кнопки переключения режимов редактирования и навигации (рис. 1).

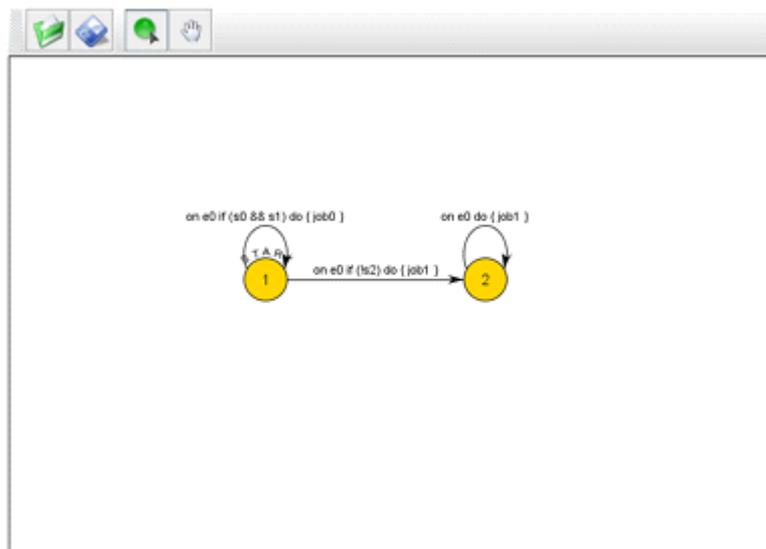


Рис. 1. Окно редактора графов переходов

1.1. Режим редактирования

Кнопка «Редактирование» (рис. 2) позволяет перейти к редактированию графа, которое выполняется при помощи мыши и клавиатуры.



Рис. 2. Режим редактирования

Ниже перечислены основные действия пользователя и их результаты (табл. 1).

Таблица 1. Основные действия при редактировании

Действия	Результат
Одиночный клик мышью в свободной части области редактирования	Добавление новой вершины
Одиночный клик по вершине, протягивание ребра с последующим отпусканием мыши над другой вершиной	Добавление нового ребра
Одиночный клик по вершине, протягивание ребра за пределы вершины с последующим возвратом и отпусканием мыши над той же вершиной	Добавление петли
Двойной клик по элементу графа	Открытие окна свойств соответствующего элемента
Одиночный клик по элементу графа (вершине, ребру или петле)	Выделение соответствующего элемента
Нажатие клавиши <i>Delete</i>	Удаление выделенных элементов

1.2. Режим навигации

Нажатие кнопки «*Навигация*» позволяет перейти в режим навигации (рис. 3). Действия по редактированию в этом режиме не доступны. Мышью можно перетаскивать изображение в разные стороны.



Рис. 3. Переключение в режим навигации

1.3. Сохранение и загрузка графов

Нажатие на кнопку «*Открыть*» (рис. 4) вызывает открытие диалогового окна, позволяющего ввести *XML*-код и, таким образом, загрузить ранее сохраненный граф. Если введенный код содержит ошибки, на экране появляется соответствующее сообщение, и загрузка графа не выполняется.



Рис. 4. Кнопки для загрузки и сохранения графа

Сохранить созданный граф, можно нажав кнопку «*Сохранить*» (рис. 4) и скопировав код в появившемся диалоговом окне. Ниже приведен пример входного файла (листинг) и соответствующий ему граф (рис. 5).

Листинг

```
<?xml version="1.0" encoding="windows-1251"?>
<automata>
  <node height="40" id="0" isStart="true" name="1" width="40" x="271" y="142">
    <edge name="edge" nodeId="2"/>
    <edge name="edge" nodeId="1"/>
  </node>
  <node height="40" id="1" isStart="false" name="2" width="40" x="439" y="120">
    <edge name="edge" nodeId="1"/>
  </node>
  <node height="40" id="2" isStart="false" name="3" width="40" x="361" y="243">
    <edge name="edge" nodeId="1"/>
  </node>
</automata>
```

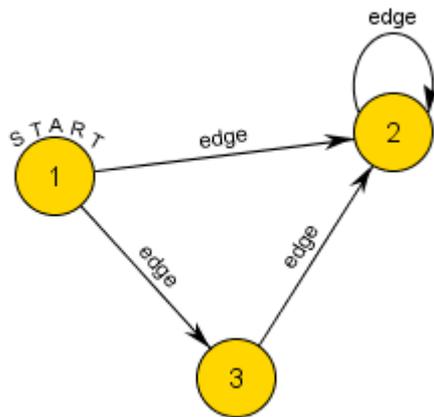


Рис. 5. Граф, соответствующий листингу

Цель игры – создать муравья «с минимальным числом состояний», который съест как можно больше яблок за число шагов, не превышающее указанное.

2.2. Исследование задачи в виртуальной лаборатории

Рассмотрим муравья, поведение которого описывается автоматом с тремя состояниями (рис. 7).

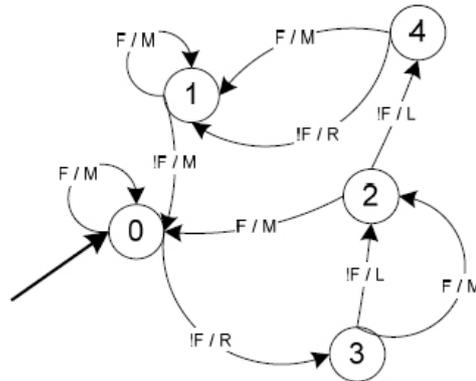


Рис. 7. Автомат с пятью состояниями

Автоматная программа, реализующая этот автомат, приведена в работе [3]. Этот же автомат можно задать, используя предлагаемый компонент (рис. 8).

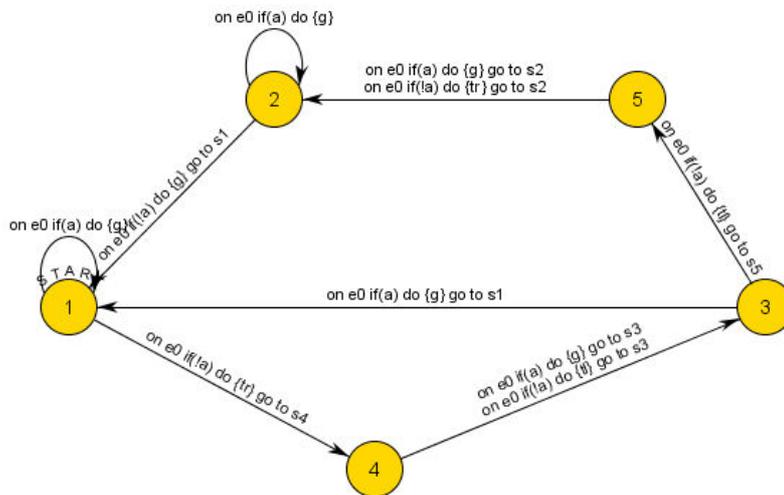


Рис. 8. Автомат с пятью состояниями

Муравей, поведение которого описывается таким автоматом, съедает только 83 яблока за 200 ходов. В работе [4] был описан автомат с семью состояниями, позволяющий муравью съесть все 89 яблок.

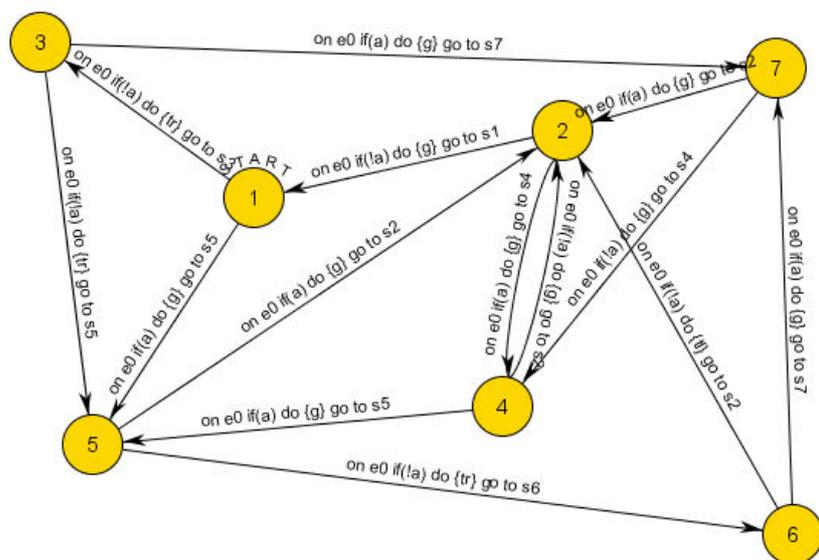


Рис. 9. Автомат, позволяющий муравью съесть всю еду

Заключение

Разработанный компонент позволяет задавать автоматы в более наглядной и естественной форме, чем с помощью текстового языка автоматного программирования. Он полностью готов для интеграции в *«Виртуальную лабораторию для первоначального обучения проектированию программ»*.

Автором была реализована минимальная функциональность, необходимая для удобной работы учащихся. В дальнейшем она может быть значительно расширена за счет добавления автоматической укладки заданных графов, построения автомата по тексту программы, генерации текста по автомату, совершенствования пользовательского интерфейса.

Источники

1. *Васильев В. Н., Казаков М. А., Корнеев Г. А и др.* Автоматное программирование и проектный подход при подготовке разработчиков программного обеспечения /Тезисы научно-технической конференции «Научное программное обеспечение в образовании и научных исследованиях». СПбГУ ПУ. 2008, с. 248–250.
http://is.ifmo.ru/download/2008-02-25_comp_proekt.pdf
2. Инструментальное средство *UniMod*.
<http://unimod.sourceforge.net/intro.html>
3. *Красильников Н.Н., Парфенов В.Г., Царев Ф.Н. и др.* Виртуальная лаборатория для первоначального обучения проектированию программ /Тезисы научно-технической конференции «Научное программное обеспечение в образовании и научных исследованиях». СПбГУ ПУ. 2008, с. 264–271.
http://is.ifmo.ru/download/2008-02-25_virtual_laboratory.pdf
4. *Царев Ф.Н., Шалыто А.А.* О построении автоматов с минимальным числом состояний для задачи об «Умном муравье» /Сборник докладов X международной конференции по мягким вычислениям и измерениям. Том 2. СПбГУ ЭТУ «ЛЭТИ». 2007, с. 88–91.
http://is.ifmo.ru/download/ant_ga_min_number_of_state.pdf
5. Виртуальная лаборатория для первоначального обучения проектированию программ. <http://rain.ifmo.ru/~krasilnikov>