

УДК 004.5

**КОНЕЧНО-АВТОМАТНАЯ РЕАЛИЗАЦИЯ ПОЛЬЗОВАТЕЛЬСКОГО  
ИНТЕРФЕЙСА ВСТРАИВАЕМОЙ СИСТЕМЫ НА ОСНОВЕ  
АСПЕКТНО-ОРИЕНТИРОВАННОГО ОПИСАНИЯ**

**В.М. Васюков**

Предлагается методика перехода от описания пользовательского интерфейса встраиваемой системы с использованием аспектно-ориентированного подхода к описанию в виде конечного автомата. Применимость предлагаемого способа демонстрируется на примере устройства индикации какой-либо физической величины. Производится описание последовательных переходов от первоначального описания пользовательского интерфейса в терминах аспектно-ориентированного подхода к ориентированному графу, описывающему возможные варианты изменения состояний пользовательского интерфейса, и от ориентированного графа к детерминированному конечному автомату, математическое описание которого дает возможность использовать его как каркас будущего прототипа пользовательского интерфейса. Рассмотренный в работе порядок перехода от описания пользовательского интерфейса встраиваемой системы в терминах аспектно-ориентированного подхода к описанию пользовательского интерфейса встраиваемой

системы в виде конечного автомата демонстрирует применимость аспектно-ориентированного подхода для разработки и прототипирования пользовательских интерфейсов встраиваемых систем.

**Ключевые слова:** аспектно-ориентированный подход, интерфейс, конечный автомат.

### Введение

В аспектно-ориентированном подходе, предложенном широкой общественности в 1998 г. [1] как новая парадигма программирования, предлагается проводить декомпозицию предметной области по сквозной функциональности. В качестве примеров использования данного подхода часто приводятся задачи ведения журналов выполнения программы или контроля предоставления доступа к ресурсам, отмечая, что данный вариант декомпозиции позволяет уменьшить сложность реализации решаемой задачи. При рассмотрении пользовательских интерфейсов встраиваемых систем можно выделить сквозную функциональность, связанную с различной реакцией элементов пользовательского интерфейса в зависимости от состояния системы. В настоящей работе рассматривается возможность описания пользовательского интерфейса в терминах аспектно-ориентированного подхода целиком, с последующим переходом к конечному автомату, отслеживающему изменение режимов работы пользовательского интерфейса.

### Описание пользовательского интерфейса в терминах аспектно-ориентированного подхода

Рассмотрим воздействие пользователя на элемент интерфейса как единицу сквозной функциональности  $a$ , а текущее состояние системы – как точку среза  $p$ . Для аспекта характерно наличие нескольких точек среза, т.е. какого-то непустого их множества  $a = \{p_1, p_2, \dots, p_n\} = \{P\}$ . При таком подходе пользовательский интерфейс можно представить как сумму аспектов  $A = \sum_{i=1}^m a_i$ , множество срезов которой может быть получено аналогичным способом:  $P_{\text{системы}} = \{P_1 \cup P_2 \cup \dots \cup P_m\}$ . Полное множество точек срезов  $P_{\text{системы}}$  включает в себя подмножества  $p_1^1, p_1^2, \dots, p_1^n$  точек срезов конкретных аспектов  $a_1, a_2, \dots, a_m$  и позволяет получить полное описание состояний интерфейса встраиваемой системы. Если какой-либо аспект  $a_i$  не имеет в своем подмножестве  $p_1, p_2, \dots, p_n$  точек среза точку из множества  $P_{\text{системы}}$  точек срезов, его поведение в данной точке описывается идентично одной из имеющихся  $p_1, p_2, \dots, p_n$  точек среза.

В качестве примера опишем как сумму аспектов  $A$  устройство индикации какой-либо физической величины с возможностью задания трех диапазонов (норма, предупреждение, авария). Описываемое устройство имеет три кнопки и четырехразрядный индикатор (рис. 1). Воздействие пользователя на систему возможно через нажатие одной из трех имеющихся кнопок. Описание аспектов, соответствующих нажатию кнопок, представлено в табл. 1. В зависимости от режима системы будет меняться и выводимая на индикатор величина. Описание аспекта «вывод» представлено в табл. 2.

Аспект		Точка среза (состояние системы / режим)	Поведение элемента
Нажатие кнопки «1»	$p_1^1$	Изменение порога предупреждения	Уменьшение значения порога предупреждения
	$p_1^2$	Изменение порога аварии	Уменьшение значения порога аварии
	$p_1^3$	Выбор для изменения порога предупреждения	Переход к $p_{11}$
	$p_1^4$	Выбор для изменения порога аварии	Переход к $p_{12}$
Нажатие кнопки «2»	$p_2^1$	Изменение порога предупреждения	Увеличение значения порога предупреждения
	$p_2^2$	Изменение порога аварии	Увеличение значения порога аварии
Нажатие кнопки «3»	$p_3^1$	Изменение порога предупреждения	Внесение изменений и переход к $p_{35}$
	$p_3^2$	Изменение порога аварии	Внесение изменений и переход к $p_{35}$
	$p_3^3$	Выбор для изменения порога предупреждения	Переход к $p_{34}$
	$p_3^4$	Выбор для изменения порога аварии	Переход к $p_{35}$
	$p_3^5$	Индикация значения измененной величины	Переход к $p_{33}$

Таблица 1. Описание нажатий кнопок

Аспект		Точка среза (состояние системы / режим)	Поведение элемента
Вывод	$p_4^1$	Изменение порога предупреждения	Вывод порога предупреждения
	$p_4^2$	Изменение порога аварии	Вывод порога аварии
	$p_4^3$	Выбор для изменения порога предупреждения	Вывод порога предупреждения
	$p_4^4$	Выбор для изменения порога аварии	Вывод порога аварии
	$p_4^5$	Индикация значения измеренной величины	Вывод измеренной величины

Таблица 2. Описание работы индикатора

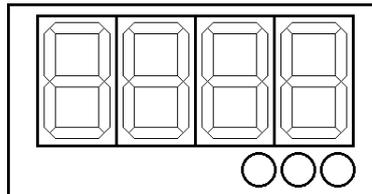


Рис. 1. Внешний вид устройства

В табл. 1, 2 индексами для элементов  $p_i^j$  служат  $i$  – номер аспекта;  $j$  – номер состояния по порядку. Для различных аспектов  $i$  состояния  $j$  с одинаковыми номерами тождественны. Получим множество срезов:

$$p_1^1 \cup p_1^2 \cup p_1^3 \cup p_1^4 \cup p_2^1 \cup p_2^2 \cup p_3^1 \cup p_3^2 \cup p_3^3 \cup p_3^4 \cup p_3^5 \cup p_4^1 \cup p_4^2 \cup p_4^3 \cup p_4^4 \cup p_4^5 = \{p_1^1 \cup p_1^2 \cup p_1^3 \cup p_1^4 \cup p_3^5\} = P_{\text{системы}}$$

**Описание ориентированного графа**

Полученное описание позволяет представить порядок изменения режимов интерфейса в виде орграфа. Формальное определение орграфа [2] выглядит как  $D = (V, E)$ , где  $E$  есть множество упорядоченных пар вершин  $v \in V$ . Так как интерфейс встраиваемой системы изменяет свое состояние (режим) в определенном порядке, то можно представить переходы между элементами множества  $P_{\text{системы}}$  как матрицу смежности  $M$  с размерностью, определяемой кардинальным числом [3],  $|P_{\text{системы}}|$ . Используя матрицу смежности  $M$ , можно построить множество  $E$ , а множеством  $V$  будет являться  $P_{\text{системы}}$ .

Создадим матрицу смежности  $M$ , размерностью  $|P_{\text{системы}}| = 5$  для нашего примера (табл. 3), в качестве ребер ориентированного графа будем использовать информацию о переходах между состояниями из табл. 1:

	$p_1^1$	$p_1^2$	$p_1^3$	$p_1^4$	$p_3^5$
$p_1^1$	–	–	1	–	–
$p_1^2$	–	–	–	1	–
$p_1^3$	–	–	–	–	1
$p_1^4$	–	–	1	–	–
$p_3^5$	1	1	–	1	–

Таблица 3. Матрица смежности  $M$

Таким образом, рассматривая пользовательский интерфейс встраиваемой системы как сумму аспектов  $A$ , можно построить ориентированный граф состояний системы (рис. 2).

**Описание конечного автомата**

Опишем пользовательский интерфейс как детерминированный конечный автомат [4] с помощью диаграммы состояний (рис. 3):

- вершины – состояния конечного автомата, в данном примере они определены на множестве  $P_{\text{системы}}$  как состояния системы  $p_i^j$ .
- ребра – переходы из одного состояния  $p_k^m$  в другое  $p_s^l$ , определены как матрица смежности  $M$  размерностью  $|P_{\text{системы}}| = 5$ ;
- нагрузка – символы, при которых осуществляется данный переход, в случае если переход из состояния  $p_m$  в состояние  $p_k$  возможен при появлении одного из нескольких символов, то над дугой диаграммы (ветвью графа) должны быть написаны они все; нагрузка описывается множеством воздействий, осуществляемых пользователем, в данном примере это нажатия на кнопки, описываемые как аспекты  $a1, a2, a3$ , эти же символы будут использоваться для обозначения осуществления переходов.

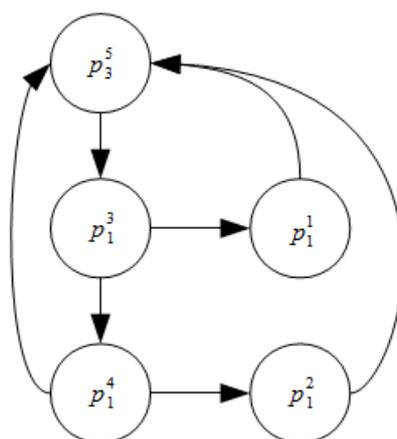


Рис. 2. Ориентированный граф

Полученное математическое описание конечного автомата дает возможность использовать его либо как каркас будущего прототипа пользовательского интерфейса, либо, при сохранении в каком-либо виде, как описание поведения пользовательского интерфейса для интерпретатора. Применение интерпретатора дает возможность использовать единый вид описания конечного автомата как для различных операционных систем, так и для различных микроконтроллеров, что позволяет применять эволюционное прототипирование, с каждой итерацией разработки приближая прототип к реальной системе.

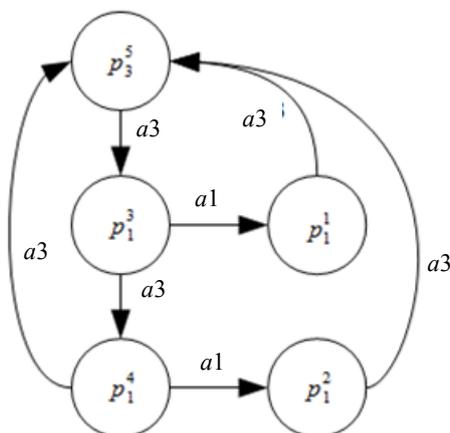


Рис. 3. Диаграмма состояний конечного автомата

### Заключение

Рассмотренная в данной работе методика перехода от описания пользовательского интерфейса встраиваемой системы в терминах аспектно-ориентированного подхода к описанию пользовательского интерфейса встраиваемой системы в виде конечного автомата демонстрирует применимость данной методики для разработки и прототипирования пользовательских интерфейсов встраиваемых систем. Использование аспектно-ориентированного подхода в рамках встраиваемых систем, характеризующихся большим числом состояний, позволяет разработчику сконцентрироваться на описании поведения элементов пользовательского интерфейса в конкретных состояниях, скрывая от него механизм контроля состояний, что снижает трудоемкость реализации прототипа.

### Литература

1. Kiczales G., Lamping J., Mendhekar A., etc. Aspect-oriented programming // Proceedings of the European Conference on Object-Oriented Programming (ECOOP). – Finland, Springer-Verlag LNCS 1241. – June 1997. – P. 220–242.
2. Кристофидес Н. Теория графов. Алгоритмический подход. – М.: Мир, 1978. – 432 с.
3. Куратовский К., Мостовский А. Теория множеств. – М.: Мир, 1970. – 416 с.
4. Хопкрофт Дж., Мотвани Р., Ульман Дж. Введение в теорию автоматов, языков и вычислений. – М.: Вильямс, 2002. – 528 с.

*Васюков Василий Михайлович* – Балтийский государственный технический университет «Военмех» им. Д.Ф. Устинова, аспирант, witcher237@rambler.ru