

ИЕРАРХИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ СОБЫТИЙНЫХ ГРАФОВ

Е. А. Бабкин (Курск)

Событийные графы являются удобным средством описания поведения дискретной системы (ДС) [1, 2, 3]. Однако недостатком событийно-ориентированного подхода считается невозможность многоуровневого иерархического представления модели и ее программной реализации [4]. Поскольку событие определяется как мгновенное изменение состояния системы или внешней среды, можно выполнять спецификации либо на уровне элементарных событий, либо на уровне макрособытий [3]. Объединение наборов событий в макрособытия имеет следующие ограничения [3]: события должны быть связаны и не должно быть задержек между ними, т.е. следование событий должно быть мгновенно. В связи с этим в работах по событийным графам рассматривается либо только одноуровневое [1], либо двухуровневое [3] событийное представление модели и одноуровневая ее программная реализация. Для процессо-ориентированного представления таких ограничений нет [4]. Поэтому в данной работе рассмотрим возможности неограниченного по числу уровней иерархического представления событийных моделей за счет расширения событийных графов введением дуг с метками спецификаций и использования иерархического представления процессов и активностей.

Основным видом вершин в событийном графе являются вершины-события. В форме графа, рассматриваемой в работах [2, 3], используются также вспомогательные вершины, отображающие следование событий: условные, распараллеливания процесса и окончания распараллеливания. Дуги в этой форме возможны двух видов: сплошные – мгновенного следования (дуги первого типа), и пунктирные – следования с задержкой (второго типа). Кроме того, используются дуги отмены событий. При непосредственном следовании событий e_i и e_j друг за другом в событийном графе это представляется либо дугой первого типа (рис. 1, а) при мгновенном следовании событий, либо дугой второго типа (рис. 1, б) при следовании событий с задержкой. В первом случае время следования равно нулю ($T_{ij} = 0$). Во втором случае время следования T_{ij} может быть определено как детерминированная или стохастическая функция.

Рис. 1. Непосредственное следование событий e_i и e_j

Введем помеченную дугу для отображения транзитивной зависимости следования событий. При иерархическом подходе к разработке и представлению модели такая дуга позволяет уточнять зависимости следования на следующем более низком уровне представления модели. На следующем более низком уровне представления модели эта зависимость детализируется. На рис. 2 представлены два случая следования событий. В первом случае (рис. 2, а) метка $\{M_{ij}\}$ указывает на дополнительную спецификацию условия следования. Во втором случае (рис. 2, б) метка $\{M_{ij}\}$ указывает на дополнительную спецификацию и условия времени следования.

Рис. 2. Следование событий e_i и e_j

Дуга второго типа, помеченная меткой спецификации $\{M_{ij}\}$, представляет на данном уровне сложную активность с событиями начала e_i и конца e_j активности, которая на следующем более низком уровне представляется процессом и описывается событийным графом. Причем процесс, детализирующий активность, имеет те же события начала e_i и конца e_j процесса, что и детализируемая активность. Таким образом, помеченная дуга отображает сложные активности-процессы, специфицируемые отдельными событийными графами.

Рассмотрим использование помеченных дуг для иерархического представления событийных графов на примере системы массового обслуживания (СМО) [2]. Транзакты поступают из источника I – входной поток N_1 . СМО состоит из канала K и накопителя H . Емкость накопителя ограничена, поэтому с накопителем связан выходной поток транзактов N_3 , получивших отказ в обслуживании в связи с конечностью соответствующего накопителя. N_2 – выходной поток транзактов, обслуженных в СМО.

Иерархия моделей функционирования определяется иерархией статических объектов и, соответственно, процессов их функционирования, существующей в рассматриваемой ДС. Процесс представляет собой связанную совокупность активностей, то есть элементарных действий, неделимых на данном уровне представления модели. С другой стороны, процессы можно рассматривать как взаимосвязанные совокупности событий. В связи с этим в соответствии с иерархией процессов множество событий модели разбивается на группы событий, специфицирующих соответствующие процессы. Спецификация имитационной модели в целом состоит из следующих основных компонентов, имеющих ссылки друг на друга: событийных графов и фрагментов текста, описывающих события и макрособытия.

На рис. 3 приведена часть спецификации верхнего уровня представления имитационной модели в виде событийного графа.

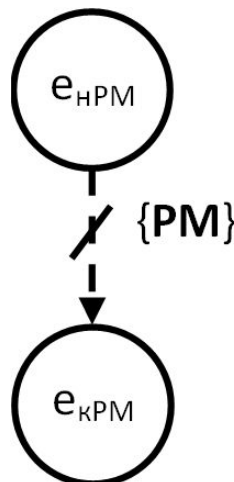


Рис. 3. Спецификация верхнего уровня представления имитационной модели

Процесс имитационного моделирования PM состоит из ожиданий поступления транзактов и из активностей a обслуживания этих транзактов. Поэтому процесс моделирования представим в виде простейшего процесса, состоящего из одной активности PM . На этом графе: $e_{нPM}$ – событие начала активности (процесса) имитационного моделирования PM , $e_{кPM}$ – событие конца активности (процесса) имитационного моделирования PM , $\{PM\}$ – метка спецификации процесса PM .

Процесс имитационного моделирования, содержащий одну сложную активность PM , детализируем на следующем событийном графе, представляющем ожидания транзактов и активности обслуживания транзактов в СМО (рис. 4). На этом графе детализируется активность PM и, соответственно, причинно-следственная зависимость между событиями $e_{нPM}$ и $e_{кPM}$. Здесь: T_{n1} – время ожидания поступления первого транзакта; T_n – время ожидания поступления следующего транзакта; i – число транзактов, обслуженных в СМО; K – число транзактов, которое должно быть обслужено в СМО, определяет длительность имитационного эксперимента; P – процесс обслуживания транзакта в СМО; A_1 – начало параллельного выполнения процесса; $e_{нP}$ – событие начала активности обслуживания транзакта в СМО; $e_{кPN2}$ – событие конца активности обслуживания транзакта в СМО (выходной поток N_2); P_4 – условие окончания процесса моделирования (по числу заявок $i = K$).

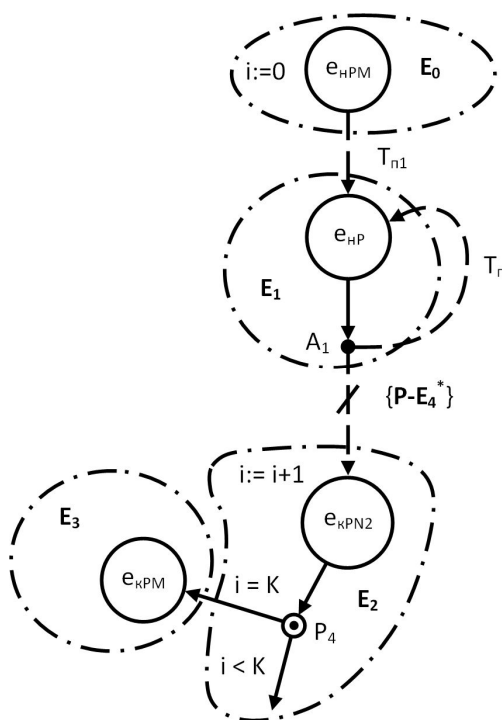


Рис. 4. Спецификация процесса моделирования PM

В событийном графе на основе подхода, изложенного в [5], выделяются следующие макрособытия: E_0 – начало процесса моделирования; E_1 – начало активности обслуживания транзакта в СМО; E_2 – конец активности обслуживания транзакта в СМО; E_3 – конец процесса моделирования; E_4^* – группа событий процесса P ; $\{P-E_4^*\}$ – метка спецификации активности (процесса) обслуживания P .

Активность обслуживания P является сложной активностью и может быть детализирована в виде процесса, состоящего из ожидания в накопителе H и обслуживания в канале K . На рис. 5 приведена спецификация процесса P обслуживания транзакта в СМО в виде событийного графа, которая детализирует транзитивную зависимость между событиями $e_{нP}$ и $e_{кP}$.

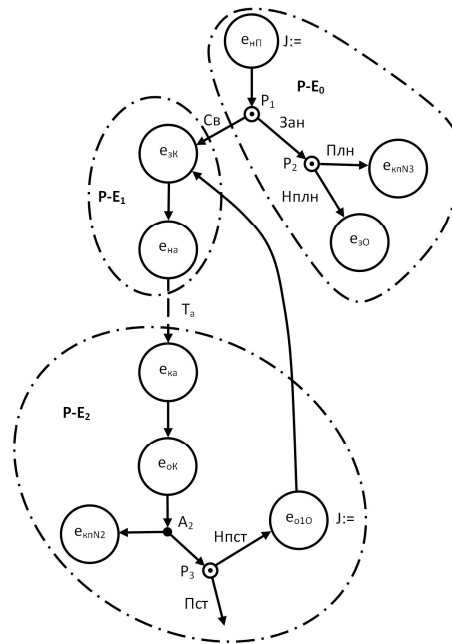


Рис. 5. Спецификация процесса обслуживания транзакта P

На этом графе: P_1 – анализ состояния z_K канала K : Св – свободен ($z_K = 0$), Зан – занят ($z_K = 1$); P_2 – анализ состояния z_H накопителя H : Нплн – неполон ($z_H < L$), Плн – полон ($z_H = L$); P_3 – анализ состояния z_H накопителя H : Нпст – непуст ($z_H > 0$), Пст – пуст ($z_H = 0$); A_2 – начало параллельного выполнения процесса; $e_{зк}$ – событие занятия канала K ; $e_{ок}$ – событие освобождения канала; $e_{на}$ – событие начала выполнения активности обслуживания a в канале K ; $e_{ка}$ – событие конца выполнения активности обслуживания a в канале K ; T_a – время выполнения активности обслуживания a в канале K ; $e_{кпN3}$ – событие конца процесса N_3 в связи с переполнением накопителя; $e_{з0}$ – событие занятия накопителя H ; $e_{о10}$ – событие освобождения из очереди накопителя H транзакта, стоящего первым. Активность обслуживания a является простой активностью. В событийном графе выделены следующие макрособытия: $P-E_0$ – анализ состояния канала и накопителя СМО; $P-E_1$ – занятие канала K ; $P-E_2$ – конец обслуживания в канале K транзакта J .

Поскольку в спецификации процесса P нет дуг с метками других спецификаций – это конечная спецификация, являющаяся нижним уровнем в представлении модели.

Спецификации процессов составляют иерархию спецификаций событийной модели. Для рассматриваемой простой модели выделены два уровня представления модели PM и P (рис. 4 и 5). Для сетей массового обслуживания число уровней будет не меньше трех: уровень имитационной модели – спецификация процесса имитационного моделирования, уровень модели сети массового обслуживания – спецификация процесса обслуживания в сети массового обслуживания, представляет порядок и маршрут обслуживания, элементами модели являются активности и события обслуживания в СМО; уровень модели СМО – спецификация процесса обслуживания в СМО, представляет порядок обслуживания в СМО, элементами модели являются активность обслуживания в канале и события обслуживания в канале и накопителе СМО. Для сложных систем уровень сетей массового обслуживания может разделяться на несколько подуровней. Событийная модель уровня СМО может быть также представлена в виде композиции моделей следующего уровня – уровня статических объектов СМО: канала и накопителя [6].

Программно-реализуемой формой событийного графа является макрособытийный граф [5]. Макрособытийный граф представляет модель на уровне макрособытий, которые реализуются в виде программных секций или процедур. Для каждого процесса событийный граф преобразуется в макрособытийный на основе метода структурной декомпозиции. В результате этого спецификацию модели можно представить также многоуровневой совокупностью спецификаций макрособытийных графов.

Предлагаемое расширение событийных графов введением дуг с метками спецификаций, представляющих транзитивные зависимости следования событий, позволяет описывать многоуровневые событийные модели ДС. Событийные модели, построенные на основе предлагаемого подхода, могут быть реализованы как программными процессорно-ориентированными, так и программными событийно-ориентированными моделями. Группирование процедур событий в спецификации программной модели в соответствии с принадлежностью выделенным процессам позволяет в значительной степени структурировать спецификацию программной модели и упростить внесение изменений в спецификацию.

Литература

1. **Schruben L. W.** Simulation Modeling with Event Graphs / L. W. Schruben // Communications of the ACM. 1983. Vol. 26. Num. 11. P. 957–963.
2. **Бабкин Е. А.** Методические указания по моделированию вычислительных систем на событийно-ориентированном языке. Курск : КПИ, 1988.
3. **Бабкин Е. А.** Событийные модели дискретных систем. Курск, 2005. 18 с. Деп. в ВИНТИ 14.01.05, № 30–В2005.
4. Автоматизация проектирования вычислительных систем. Языки, моделирование и базы данных / Под ред. М. Брейера. М.: Мир, 1979. С. 12–29, 35–44.
5. **Бабкин Е. А.** О синтезе событийных моделей дискретных систем // Ученые записки : электронный научный журнал Курского государственного университета. Эл № 77-26463. 2006. № 1. 17 с. <http://www.scientific-notes.ru/pdf/s15.pdf>.
6. **Бабкин Е. А.** О методах декомпозиции событийных графов. / Е. А. Бабкин, Е. А. Бобрышев // Ученые записки: электронный научный журнал Курского государственного университета, № гос. регистрации №0420800068\0024. 2008. № 2(6). 17 с. <http://www.scientific-notes.ru/pdf/006-02.pdf>.