УДК 004.514.2

ФОРМАЛЬНОЕ ПРЕДСТАВЛЕНИЕ СОБЫТИЙНЫХ МОДЕЛЕЙ ДИСКРЕТНЫХ СИСТЕМ

© 2012 Е. А. Бабкин[1], В. В. Разиньков[2]

[1]канд. техн. наук, доцент, профессор каф. программного обеспечения и администрирования информационных систем
e-mail: eababkin@gmail.com
[2]аспирант каф. программного обеспечения и администрирования информационных систем
e-mail: razinkov86@gmail.com

Курский государственный университет

Вводятся понятия и символьные обозначения отношений следования и отмены событий, потоков событий и операций соединения и разделения потоков. Рассматривается формальное символьное представление событийного поведения дискретных систем на уровне элементарных событий и макрособытий.

Ключевые слова: событие, событийный граф, событийное моделирование, отношение следования событий, отношение отмены событий, поток событий.

Введение

Практическими источниками развития теории имитационного событийного моделирования дискретных систем являются событийные языки моделирования, такие как SIMSCRIPT [Kiviat и др. 1969], SMPL [Макдугал М. 1979] и СЛАМ [Прицкер 1987]. Первыми теоретическими работами в рассматриваемой области стали работы M.R. Lackner [1964; 1962], в которых он представил теорию дискретных событийных систем. В этих работах изменение, а не время стало первоначальным. Предложенная M.R. Lackner теория исчисления изменений требует, чтобы время определялось в терминах изменений. L. Schruben [1983] предложил графическое представление событийной модели в виде событийного графа (event graphs), отображающего не состояния системы, а ее изменения. В дальнейшем предлагались некоторые изменения в событийных графах, однако, работа L. Schruben осталась наиболее заметной. В работах Е.А. Бабкина [1988; 2005] было предложено другое представление событийного графа, отличающееся способом графического представления причинно-следственных зависимостей и методологией выделения событий и построения событийного графа.

Одной из проблем событийного моделирования является то, что одно и то же событийное поведение дискретной системы может быть представлено различными событийными графами, которые могут быть программно реализованы с различной эффективностью. E. Yücesan [1989] показал, что используемые подходы могут приводить к ошибочным результатам, так как событийный граф неустойчив к влиянию мировоззрений исследователя. Для одной и той же системы разные исследователи могут построить различные событийные графы. В связи с этим возникает задача формализации эквивалентных преобразований, анализа и синтеза событийных графов. Для этого удобно использовать символьное представление событийной модели дискретной системы.

Отношения следования и отмены событий

Можно выделить два вида зависимостей между событиями: причинно-следственную и информационную.

В работе Kranzlmuller и др. [1996] событийный граф определяется как ориентированный граф $G = (E, \rightarrow)$, где E – непустое множество событий e графа G, а $\rightarrow$ – отношение, соединяющее события, так что $x \rightarrow y$ означает, что есть ребро из события x до события y в G с началом в событии x и концом в событии y. В этой работе вводится обозначение отношения следования между событиями в событийном графе в виде символа $\rightarrow$. Однако нет отображения важных параметров отношения следования, и прежде всего параметра времени. Кроме того, не рассматриваются условия следования и символьное представление отношения отмены событий.

В работах Бабкина, Бобрышева [2007; 2007а] вводится ***отношение следования*** (***планирования***) ***событий***, отображаемое стрелкой $\rightarrow^{\Delta t}$. Это бинарное отношение описывает логические ситуации «предыдущий–следующий» и отображается как

$$\alpha \rightarrow^{\Delta t} e,$$

здесь α – высказывание о происхождении события или некоторая логическая функция высказываний об инициирующих событиях и состоянии системы;

e – планируемое событие;

Δt – время задержки следования (планирования), то есть время, через которое должно наступить событие e, после того как функция α примет истинное значение.

Функция α является ***событийной логической функцией***, то есть функцией, принимающей значение «истинно», только в момент совершения событий [Бабкин 2010].

Если после события e_1 через время Δt следует событие e_2 при условии истинности c, то это в символьной форме можно представить следующим образом:

$$e_1 \wedge c \rightarrow^{\Delta t} e_2 .$$

Событие e_2 обязательно произойдет только при условии, если произошло событие e_1 и в этот момент логическое выражение c было истинно. Говорят также в этом случае, что событие e_1 планирует событие e_2 через время Δt при условии c или, что то же самое, событие e_2 следует за событием e_1 при условии c через время Δt:

$$e_2^{\Delta t} \leftarrow e_1 \wedge c .$$

Событие e_1 в этом случае будем называть ***событием-предшественником***, а событие e_2 – ***событием-последователем***.

Следует заметить, что событие e_2 может планироваться несколькими событиями или же планироваться при условии, что произошли или не произошли некоторые события. В этом случае в событийной логической функции α допустимы отношения дизъюнкции, конъюнкции или отрицания, включающие события. Например, если событие e_3 следует через время Δt после события e_1 или после события e_2 при выполнении условия c в момент времени, когда происходят события e_1 или e_2, то это в символьной форме можно представить следующим образом:

$$e_3^{\Delta t} \leftarrow (e_1 \vee e_2) \wedge c ,$$

или, что то же самое, в эквивалентной форме:

$$(e_1 \vee e_2) \wedge c \rightarrow^{\Delta t} e_3 .$$

Выделим следующие виды отношений следования событий: безусловное, условное, без задержки во времени (мгновенное), с задержкой, с неизвестной задержкой и с необязательной задержкой.

Мгновенное безусловное следование события e_2 после события e_1 будем в символьном виде представлять как

$$e_1 \rightarrow^0 e_2 ,$$

безусловное следование события e_2 после события e_1 через время Δt –

$$e_1 \rightarrow^{\Delta t} e_2 ,$$

условное следование после события e_1, при выполнении в этот момент условия c, через время Δt события e_2 –

$$e_1 \wedge c \rightarrow^{\Delta t} e_2 .$$

Необходимо отметить, что это случай следования событий с предусловием.

Следование с постусловием требует ввода фиктивного (пустого) события $e_\varnothing$, то есть события, которому в системе не сопоставлено никакого изменения. Это событие может означать, например, факт наступления запланированного момента времени. Следование с постусловием c можно представить следующим образом:

$$e_1 \rightarrow^{\Delta t} e_\varnothing \wedge c \rightarrow^0 e_2 .$$

Это означает, что после события e_1, происходящего в момент времени t_1, через время Δt происходит событие e_2, если в этот момент $t_1 + \Delta t$ выполняется условие c.

Безусловное следование события e_2 после события e_1 через неизвестное время, которое необходимо в дальнейшем определить, будем обозначать как

$$e_1 \rightarrow^? e_2 .$$

Такое обозначение может быть удобно на начальных этапах формализации модели.

Безусловное следование события e_2 после события e_1, когда нас не интересует время следования, будем обозначать как

$$e_1 \rightarrow e_2 .$$

Введем ***отношение отмены событий***, отображаемое стрелкой $\rightarrow^{\Delta t}_{cancel}$, где *cancel* – признак отношения отмены, знак Δt – время задержки отмены события. Это отношение описывает класс пар логических ситуаций «предыдущий–следующий» и отображается как

$$\alpha \rightarrow^{\Delta t}_{cancel} e .$$

Если событие e_1 через время Δt отменяет событие e_2 при условии истинности c, то это в символьной форме можно представить следующим образом:

$$e_1 \wedge c \rightarrow^{\Delta t}_{cancel} e_2 .$$

Совершение события e_2 обязательно будет отменено только при условии, если это событие было ранее запланировано, произошло событие e_1 и в этот момент логическое выражение c было истинно. Говорят также в этом случае, что событие e_2 отменяется событием e_1:

$$e_2 {}^{\Delta t}_{cancel} \leftarrow e_1 \wedge c .$$

Виды отношений отмены событий такие же, как и виды отношений следования.

Мгновенную безусловную отмену события e_2 после события e_1 будем представлять как

$$e_1 \rightarrow^{0}_{cancel} e_2 ,$$

безусловную отмену события e_2 после события e_1 через время Δt как

$$e_1 \rightarrow^{\Delta t}_{cancel} e_2 ,$$

условную отмену события e_2 после события e_1 через время Δt, если выполняется условие c, как

$$e_1 \wedge c \rightarrow^{\Delta t}_{cancel} e_2 ,$$

безусловную отмену события e_2 после события e_1 через неизвестное время, которое необходимо в дальнейшем определить, как

$$e_1 \rightarrow^{?}_{cancel} e_2 ,$$

безусловную отмену события e_2 после события e_1, когда нас не интересует время задержки отмены, как

$$e_1 \rightarrow_{cancel} e_2 .$$

Таким образом, событие e_2 ***причинно-следственно зависит*** от события e_1, если e_2 связано с e_1 отношением следования или отмены. Два события ***причинно-следственно независимы***, если между ними нет отношений следования и отмены.

Основываясь на классификациях переменных состояния и конфликтов действий в модели, данных в работе [Page 1999], определим информационную зависимость событий. Два события e_1 и e_2 ***информационно зависимы***, если

$$(O_1 \cap O_2) \cup (I_1 \cap O_2) \cup (I_2 \cap O_1) \neq \varnothing ,$$

где O_1 – подмножество переменных состояния, изменяемых при наступлении события e_1,

O_2 – подмножество переменных состояния, изменяемых при наступлении события e_2,

I_1 – подмножество переменных состояния, влияющих на наступление события e_1,

I_2 – подмножество переменных состояния, влияющих на наступление события e_2.

Подмножество I_1 – это подмножество переменных, входящих во множество выражений условий следования $C_1 = \{c_i^1\}$ события e_1 после всех событий-предшественников и во множество выражений функций изменения $F_1 = \{f_i^1\}$ состояний переменных, то есть

$$I_1 = I_{C1} \cup I_{F1} .$$

Аналогично для подмножества I_2

$$I_2 = I_{C2} \cup I_{F2} .$$

Два события e_1 и e_2 ***информационно независимы***, если

$$(O_1 \cap O_2) \cup (I_1 \cap O_2) \cup (I_2 \cap O_1) = \varnothing .$$

Два события e_1 и e_2 ***независимы***, если они независимы причинно-следственно и информационно одновременно. Независимые события могут происходить как последовательно, так и одновременно. Зависимые причинно-следственно события могут происходить только последовательно. Ограничения на одновременность наступления информационно зависимых событий определяются только информационными зависимостями в момент совершения событий. В этом случае необходимо рассматривать предусловия и постусловия событий, истинность которых вычисляется в момент совершения событий. Допустимость одновременного выполнения экземпляров одного события зависит от характера (природы) моделируемых процессов.

Два независимых события e_1 и e_2, происходящие в моменты времени t_1 и t_2 соответственно, являются одновременными, если $t_1 = t_2$.

Два независимых события e_1 и e_2, происходящие в моменты времени t_1 и t_2 соответственно, являются неодновременными, если $t_1 \neq t_2$, причем событие e_2 происходит после события e_1, если $t_2 > t_1$.

Два зависимых события e_1 и e_2, связанные отношением следования $e_1 \wedge c \rightarrow^{\Delta t} e_2$, являются неодновременными, даже если $\Delta t = 0$. Формально в модели они происходят в один момент времени, но e_2 после e_1. Если два зависимых события e_1 и e_2 связаны отношением отмены $e_1 \wedge c \rightarrow^{\Delta t}_{cancel} e_2$ и $\Delta t = 0$, то отмена события e_2 происходит после наступления e_1 при истинности условия c.

Потоки событий

Поток событий – это детерминированная или стохастическая последовательность событий, происходящих одно за другим в какие-то моменты времени [Советов, Яковлев 2005]. Таким образом, потоком событий будем называть упорядоченное множество событий $Tr = \left\langle e_i \middle| i = \overline{1,n} \right\rangle$, здесь i – номер события в потоке.

Признаками упорядочения потоков событий являются время наступления этих событий и наличие информационной или причинно-следственной зависимости между событиями.

Рассмотрим потоки событий, упорядоченные по времени поступления этих событий.

Однородный поток событий $Tr = \left\langle e_i \middle| i = \overline{1,n} \right\rangle$ характеризуется только множеством моментов времени наступления этих событий $T_{Tr} = \left\langle t_i \middle| i = \overline{1,n} \right\rangle$, причем $t_{i-1} < t_i$. Время события e_i может задаваться интервалом t_i от некоторого одного момента времени – начального момента времени – или интервалом Δt_i между двумя следующими друг за другом событиями. То есть однородный поток событий может быть также задан в виде последовательности интервалов времени Δt_i между i-м и $(i–1)$-м событиями Tr.

Неоднородный поток событий характеризуется не только моментами наступления событий, но и параметрами событий, то есть событие может быть представлено как $e_i = \left\langle t_i, p_{1i}, ..., p_{ki} \right\rangle$, где t_i – момент времени совершения события, p_{1i}, …, p_{ki} – набор признаков события e_i. Потоки событий, для которых известно время наступления, – это наблюдаемые реализации событийного поведения систем. Например, для неоднородного потока событий может быть задано время выполнения события t_i, объект O_j, в котором происходит событие, приоритет события Pr_i и функция изменения f_i [Бабкин 2010].

Нотации событийного поведения могут быть представлены в виде совокупности потоков событий, связанных отношениями следования или отмены, для которых время не является непосредственной характеристикой, то есть событие e_i характеризуется только признаками события $p_{1i}, \ldots, p_{ki}$.

Потоком зависимых событий будем называть такое упорядоченное множество событий $Tr = \left\langle e_i \middle| i = \overline{1, n} \right\rangle$, что

$$\forall e_i \in Tr / \{e_1\} \exists! e_j \in Tr \left((e_j \wedge c_j \rightarrow^{\Delta t_j} e_i) \oplus (e_j \wedge c_j \rightarrow^{\Delta t_j}_{cancel} e_i) \right),$$

где c_j и Δt_j – соответственно условие и время задержки следования или отмены события e_i после события e_j, причем $j = i - 1$.

Поток событий, состоящий из двух событий, связанных отношением следования или отмены, будем называть ***элементарным***.

В случае описания зависимых потоков событий используется последовательная нотация их следования, например поток с условным следованием четырех событий:

$$e_1 \wedge c_1 \rightarrow^{\Delta t_1} e_2 \wedge c_2 \rightarrow^{\Delta t_2} e_3 \wedge c_3 \rightarrow^{\Delta t_3} e_4 .$$

Для уменьшения размера символьного событийного описания введем сокращенные нотации.

Последовательное безусловное следование n событий без задержки во времени

$$e_1 \rightarrow^0 e_2 \rightarrow^0 \ldots \rightarrow^0 e_n$$

будем обозначать как

$$(e_1; e_2; \ldots; e_n),$$

а одновременное следование n независимых событий e_1, e_2, ..., e_n –

$$(e_1 | e_2 | \ldots | e_n).$$

Например, одновременное безусловное следование n событий без задержки во времени после события e_0

$$e_0 \rightarrow^0 e_1,$$
$$e_0 \rightarrow^0 e_2,$$
$$\ldots$$
$$e_0 \rightarrow^0 e_n$$

будем обозначать как

$$e_0 \rightarrow^0 (e_1 | e_2 | \ldots | e_n);$$

одновременное условное следование n событий с задержкой во времени Δt после события e_0 с общим условием c

$$e_0 \wedge c \rightarrow^{\Delta t} e_1,$$
$$e_0 \wedge c \rightarrow^{\Delta t} e_2,$$
$$\ldots$$
$$e_0 \wedge c \rightarrow^{\Delta t} e_n$$

обозначим

$$e_0 \wedge c \rightarrow^{\Delta t} (e_1 | e_2 | \ldots | e_n).$$

Два потока $Tr_1 = \left\langle e_i^1 \mid i = \overline{1,n} \right\rangle$ и $Tr_2 = \left\langle e_j^2 \mid j = \overline{1,k} \right\rangle$, где верхний индекс определяет принадлежность потоку событий, назовем ***непересекающимися***, если $\forall e_i^1 \in Tr_1 \forall e_j^2 \in Tr_2 (e_i^1 \neq e_j^2)$, ***пересекающимися***, если $\exists e_i^1 \in Tr_1 \exists e_j^2 \in Tr_2 (e_i^1 = e_j^2)$.

Потоки могут быть связаны отношением следования или отмены, так же как и события. Например, если после потока $Tr_1 = \left\langle e_i^1 \mid i = \overline{1,n} \right\rangle$ безусловно следует поток $Tr_2 = \left\langle e_j^2 \mid j = \overline{1,k} \right\rangle$ и потоки не имеют общих событий, то это будет представлено как $Tr_1 \rightarrow Tr_2$, что означает: после выполнения последнего события $e_n^1 \in Tr_1$ выполняется первое событие $e_1^2 \in Tr_2$, то есть $e_n^1 \rightarrow e_1^2$.

Два наблюдаемых потока событий $Tr_1 = \left\langle e_i^1 \mid i = \overline{1,n} \right\rangle$ и $Tr_2 = \left\langle e_j^1 \mid j = \overline{1,k} \right\rangle$ являются ***последовательными***, если интервалы времени, в течение которых они происходят, не пересекаются, то есть $\left[t_1^1, t_n^1\right] \cap \left[t_1^2, t_k^2\right] = \varnothing$. Причем если $t_1^2 > t_n^1$, то после потока Tr_1 следует поток Tr_2, а если $t_k^2 < t_1^1$, то после потока Tr_2 следует поток Tr_1.

Два наблюдаемых потока событий $Tr_1 = \left\langle e_i^1 \mid i = \overline{1,n} \right\rangle$ и $Tr_2 = \left\langle e_j^1 \mid j = \overline{1,k} \right\rangle$ являются ***параллельными*** (одновременными), если интервалы времени, в течение которых они происходят, совпадают $\left[t_1^1, t_n^1\right] = \left[t_1^2, t_k^2\right]$, то есть $t_1^1 = t_1^2$ и $t_n^1 = t_k^2$, и ***частично параллельными***, если интервалы времени, в течение которых они происходят, пересекаются $\left[t_1^1, t_n^1\right] \cap \left[t_1^2, t_k^2\right] \neq \varnothing$, то есть $t_1^1 \neq t_1^2$ или $t_n^1 \neq t_k^2$.

Сокращенную нотацию последовательного безусловного следования *n* потоков без задержки во времени

$$Tr_1 \rightarrow^0 Tr_2 \rightarrow^0 \ldots \rightarrow^0 Tr_n$$

будем обозначать как

$$(Tr_1; Tr_2; \ldots; Tr_n).$$

Параллельное следование *n* потоков обозначим

$$(Tr_1 \mid Tr_2 \mid \ldots \mid Tr_n).$$

Два потока $Tr_1 = (Tr_1^1 \rightarrow Tr_2^1)$ и $Tr_2 = (Tr_1^2 \rightarrow Tr_2^2)$ сходятся, если они являются пересекающимися и $Tr_1^1 \neq Tr_1^2, Tr_2^1 = Tr_2^2$, и расходятся, если они являются пересекающимися и $Tr_1^1 = Tr_1^2, Tr_2^1 \neq Tr_2^2$.

Поток $Tr_2 = \left\langle e_j^1 \mid j = \overline{1,k} \right\rangle$ является продолжением потока $Tr_1 = \left\langle e_i^1 \mid i = \overline{1,n} \right\rangle$, если $e_n^1 = e_1^2$, то есть первое событие потока Tr_2 совпадает с последним событием потока Tr_1.

Для выполнения эквивалентных преобразований символьных нотаций потоков необходимо ввести операции соединения (*Join*) и разделения (*Part*) потоков. Возможны две разновидности этих операций: по общему событию и по отношению следования.

Операция ***соединения потоков по общему событию***:

$Tr_1 \; Join \; (e_i^1 = e_j^2) \; Tr_2 = Tr$,

где $e_i^1 \in Tr_1 \subset Tr$, $e_j^2 \in Tr_2 \subset Tr$. Если e_i^1 является последним событием потока Tr_1, а e_j^2 – первым событием потока Tr_2, то это последовательное соединение потоков Tr_1 и Tr_2. При этом события e_i^1 и e_j^2 в потоке Tr совмещаются в одно событие.

Операция ***разделения потоков по общему событию*** e_g:

$$Tr\ Part(e_g) = \langle Tr_1, Tr_2 \rangle,$$

где $e_g = e_g^1 = e_1^2$, $e_g \in Tr$, $e_g^1 \in Tr_1$, $e_1^2 \in Tr_2$. Событие e_g^1 является последним событием потока Tr_1, а e_1^2 – первым событием потока Tr_2, то есть поток Tr_2 является продолжением потока Tr_1.

Символьное представление событийных моделей

Событийное поведение дискретной системы можно представить в виде совокупности зависимых потоков событий. Причем символьную нотацию возможно использовать для описания поведения системы на уровне как элементарных, так и составных событий и макрособытий.

Рассмотрим пример системы массового обслуживания с конечным накопителем и ограничением по времени ожидания в накопителе [Бабкин, Бобрышев 2008]. Поведение такой системы может быть описано следующим набором зависимых потоков событий:

$$Tr_1 = e_{нПм} \rightarrow^{T0} e_{нП},$$

$$Tr_2 = e_{нП} \rightarrow^{Tn3} e_{нП},$$

$$Tr_3 = e_{нП} \wedge (K = Св) \rightarrow^{0} (e_{зК} \mid e_{на}),$$

$$Tr_4 = e_{нП} \wedge (K = Зан) \wedge (H = Нплн) \rightarrow^{0} e_{зН} \rightarrow^{Tн} e_{оJH},$$

$$Tr_5 = e_{нП} \wedge (K = Зан) \wedge (H = Плн) \rightarrow^{0} e_{кПN3},$$

$$Tr_6 = (e_{оJH}; e_{кПN3}),$$

$$Tr_7 = e_{на} \rightarrow^{Ta} (e_{ка}; e_{оK}),$$

$$Tr_8 = e_{оK} \rightarrow^{0} e_{кПN2} \wedge (i = N) \rightarrow^{0} e_{кПм},$$

$$Tr_9 = e_{оK} \wedge (H = Нпст) \rightarrow^{0} e_{о1H},$$

$$Tr_{10} = e_{о1H} \rightarrow^{0} (e_{зK} \mid e_{на}),$$

$$Tr_{11} = e_{о1H} \rightarrow^{0}_{cancel} e_{оJH}.$$

Потоки Tr_2, Tr_3, Tr_4 и Tr_5 являются расходящимися, поскольку имеют общее начало: событие $e_{нП}$. Кроме того, эти потоки являются продолжением потока Tr_1, поскольку Tr_1 заканчивается событием $e_{нП}$, с которого они начинаются. В связи с этим можно соединить, например, потоки Tr_1 и Tr_3:

$$Tr_{1,3} = Tr_1\ Join\ (e_{нП}^1 = e_{нП}^3)\ Tr_3 =$$

$$= e_{нПм} \rightarrow^{T0} e_{нП} \wedge (K = Св) \rightarrow^{0} (e_{зK} \mid e_{на}).$$

У потока Tr_2 начальное и конечное события совпадают, следовательно, это циклический поток. Сходящимися потоками являются потоки Tr_3 и Tr_{11}, Tr_5 и Tr_6.

Возможно символьное описание поведения системы на уровне макрособытий [Бабкин 2005]. Например, макрособытийное описание для рассматриваемого примера:

$E_0 \rightarrow^{T0} E_1$,

$E_1 \rightarrow^{Tn3} E_1$,

$E_1 \wedge (K = Св) \rightarrow^{Ta} E_2$,

$E_1 \wedge (K = Зан) \wedge (H = Нплн) \rightarrow^{Tн} E_3$,

$E_2 \wedge (H = Нпст) \rightarrow^{Ta} E_2$,

$E_2 \wedge (i = N) \rightarrow^{0} E_4$,

$E_2 \wedge (H = Нпст) \rightarrow^{0}_{cancel} E_3$.

Каждое макрособытие при этом может быть также специфицировано в символьной форме:

E_0 : макрособытие начала моделирования

$e_{нПм}$;

E_1 : макрособытие поступления транзакта и начала обслуживания транзакта в СМО

$e_{нП} \wedge (K = Св) \rightarrow^{0} (e_{зК} \mid e_{на})$,

$e_{нП} \wedge (K = Зан) \wedge (H = Нплн) \rightarrow^{0} e_{зН}$,

$e_{нП} \wedge (K = Зан) \wedge (H = Плн) \rightarrow^{0} e_{кПN3}$;

E_2 : макрособытие окончания обслуживания транзакта в СМО

$(e_{ка}; e_{оК})$,

$e_{оК} \rightarrow^{0} e_{кПN2}$,

$e_{оК} \wedge (H = Нпст) \rightarrow^{0} (e_{о1Н}; (e_{зК} \mid e_{на}))$;

E_3 : макрособытие отказа в обслуживании транзакта в СМО в связи с превышением времени ожидания в накопителе H

$(e_{оJН}; e_{кПN3})$;

E_4 : макрособытие окончания моделирования

$e_{кПм}$.

Заключение

Рассмотренные формальные символьные представления причинно-следственных отношений событий, потоков зависимых событий и операций над потоками событий в совокупности составляют алгебру зависимых событий, позволяющую в формальном виде представить поведение дискретной системы как на уровне элементарных событий, так и на уровне макрособытий. Предложенный формализм позволяет рассмотреть вопросы анализа, преобразования и синтеза событийных моделей.

Библиографический список

Бабкин Е. А. Методические указания по моделированию вычислительных систем на событийно-ориентированном языке. Курск: КПИ, 1988.

Бабкин Е. А. Событийные модели дискретных систем / Курск. гос. ун-т. Курск, 2005. 18 с. Деп. в ВИНИТИ 14.01.05, № 30–В2005.

Бабкин Е. А., Бобрышев Е. А. Иерархическое событийно-автоматное моделирование // Информационные технологии моделирования и управления. Научно-технический журнал. Воронеж: Научная книга, 2007. № 1 (35).

Бабкин Е. А., Бобрышев Е. А. О формализме событийно-автоматного моделирования // Третья всероссийская научно-практическая конференция по имитационному моделированию и его применению в науке и промышленности «Имитационное моделирование. Теория и практика» ИММОД-2007: сб. докладов. Т. 1. СПб.: ФГУП ЦНИИТС, 2007а. С. 82–86.

Бабкин Е. А., Бобрышев Е. А. О методах декомпозиции событийных графов // Ученые записки. Электронный научный журнал Курского государственного университета. Курск, 2008. № 2(6). 17 с. URL: http://scientifical-notes.ru/pdf/006-02.pdf (дата обращения: 11.02.2012). № гос. регистрации №0420800068\0024.

Бабкин Е. А. О понятии события в дискретно-событийном моделировании // Информационные системы: Теория и практика: сб. науч. работ / редкол.: Е.А. Бабкин, В.А. Кудинов, И.В. Селиванова; отв. ред. Е.А. Бабкин; фак. информатики и вычислит. техники Курск. гос. ун-та. Курск, 2010. С. 46–51.

Макдугал М. Моделирование на уровне систем // Автоматизация проектирования вычислительных систем. Языки, моделирование и базы данных / под ред. М. Брейера. М.: Мир, 1979. С. 11–133.

Прицкер А. Введение в имитационное моделирование и язык СЛАМ II: пер. с англ. М., 1987. 646 с.

Советов Б. Я., Яковлев С. А. Моделирование систем: учеб. пособие для вузов. 4 изд., стер. М.: Высшая школа, 2005. 343 с.

Kiviat P.J., Villanueva R., Markovitz H. The SIMSCRIPT II Programming Language. Prentica-Hall, 1969.

Kranzlmuller D., Grabner S., Volkert J. Program Analysis Through Visualization // Software Visualization / Eades, P., Zhang, K., (Eds.) / Series on Software Engineering and Knowledge Engineering. 1996. Vol. 7. World Scientific Publishing, Singapore. P. 183–202.

Lackner M.R. Toward a General Simulation Capability / Proceedings of the AFIPS Spring Joint Computer Conference. San Francisco, CA, May 1–3, 1962. P. 1–14.

Lackner M.R. Digital Simulation and System Theory. System Development Corporation, Santa Monica, CA, 1964.

Page E., Nance R.. Incorporating Support for Model Execution within the Condition Specification // Transactions of the Society for Computer Simulation. 1999. 16(1). P. 47–62.

Schruben L.W. Simulation Modeling with Event Graphs // Communications of the ACM. 1983. Vol. 26. №. 11. P. 957–963.

Yücesan E. SimulationGraphs for the Design and Analysis of Discrete EventSimulation Models: PhD Dissertation. Cornell University, Ithaca, NY,1989.