

УДК 004.652

Данилина Елена Александровна, студентка, Elena Alexandrovna Danilina

Королёва Дарья Андреевна, студентка, Daria Andreevna Koroleva

Шибанов Сергей Владимирович, кандидат технических наук, доцент, Sergey Vladimirovich Shibanov

Пензенский государственный университет

МОДЕЛЬНОЕ ПРЕДСТАВЛЕНИЕ СЛОЖНЫХ СОБЫТИЙ

MODEL REPRESENTATION OF COMPLEX EVENTS

Аннотация. В данной работе рассматриваются виды событий и способы их модельного представления с применением конечных автоматов и сетей Петри.

Annotation. This paper discusses the types of events and methods of their model representation using finite automata and Petri nets.

Ключевые слова: событие, журнал событий, конечный автомат, сети Петри, алгебра событий, моделирование процессов.

Keywords: event, event log, finite automata, Petri nets, event algebra, process modeling.

Все выявленные события делятся на простые и сложные (составные) события.

Простое событие – мгновенное событие, происходящее в определенный момент времени, которое не может быть разбито на составные события.

Сложные события – события, которые определяются комбинацией простых или сложных событий с применением операторов алгебры событий [1].

Для описания шаблонов сложных событий применяется следующий набор операций алгебры событий:

- последовательность ($;$): $(E1;E2);$
- дизъюнкция (\vee): $(E1 \vee E2);$
- конъюнкция (\wedge): $(E1 \wedge E2);$
- одновременно ($=$): $(E1=E2);$
- отрицание (\neg): $(E1 \wedge \neg E2);$
- любое : любое(n) $E1$, когда происходит n событий типа $E1$;
- аperiодический (Ap): $Ap(E2, E1, E3) , E2$ в пределах $E1$ и $E3$;
- периодический (Per): $Per(t, E1, E2) ,$ каждые t временных шагов между $E1$ и $E2$ [2].

Для моделирования сложных событий в процессе их обнаружения в реальном времени или в процессе их выявления на основе журнала событий, необходимо использовать модельное представление сложных событий [3].

В ходе проведенного исследования были рассмотрены такие методики модельного представления детерминированные конечные автоматы и сети Петри;

Конечный автомат — это некоторая абстрактная модель, содержащая конечное число состояний чего-либо. В один момент времени только одно

состояние может быть активным. Следовательно, для выполнения каких-либо действий машина должна менять свое состояние.

Формализованное представление автомата выглядит следующим образом:

$M = (V, Q, q_0, F, \delta)$, где

- V — входной алфавит, применительно к задаче моделирования сложных событий V - множество всех событий, которые могут произойти;
- Q — множество внутренних состояний,
- q_0 — начальное состояние ($q_0 \in Q$), для нашей задачи соответствует источнику сложного события или инициатору;
- F — множество заключительных, или конечных состояний ($F \subset Q$), в данной задаче соответствует множеству событий-терминаторов или событий, которые являются завершающими в рамках рассматриваемого сложного события;
- δ — функция переходов, в задачи моделирования сложных событий данный параметр будет соответствовать множеству операций алгебры событий.

Конечные автоматы подразделяются на детерминированные и недетерминированные.

В свою очередь, модель сети Петри состоит из четырех основных элементов: позиций, переходов, дуг и маркеров).

Сети Петри представляют собой кортеж (P, T, F) , где:

1. P — конечное множество позиций (множество событий, которые могут быть выявлены),
2. T — конечное множество переходов (множество операций алгебры событий),
3. $F \subseteq (P \times T) \cup (T \times P)$ представляет собой набор дуг, который называется отношением потока [3].

Понятию позиции в задачи моделирования сложных событий соответствует понятию событию.

У каждого перехода может быть ноль или несколько входящих дуг, идущих от своих исходных позиций, а также ноль или несколько исходящих дуг, направляющихся к выходным позициям, то есть событие может быть причиной появления другого события или, наоборот, препятствием возникновения какого-либо события.

Переход называется разрешенным, если имеется хотя бы один маркер на его входных позициях. Любой разрешенный переход может срабатывать по желанию, удаляя маркер со всех исходных позиций и размещая их на всех выходных позициях. Количество маркеров на исходных и выходных позициях не обязано совпадать. Если разрешены два или более переходов, произойти может любой из них. Благодаря недетерминированности выбора перехода, который произойдет, сети Петри могут использоваться для моделирования протоколов.

Сети Петри могут представляться в виде набора алгебраических формул, напоминающих грамматические правила. Каждому переходу

соответствует одно правило грамматики. Каждое такое правило описывает входные и выходные позиции перехода.

Для моделирования сложных событий в наибольшей степени подходят Сети Петри, так как они имеют комплекс функциональных возможностей, необходимых для моделирования цепочек сложных событий, а именно:

- возможность указать событие-инициатор, источник сложного событий, и событие-терминатор, которое являлось завершением рассматриваемой цепочки сложного события;

- так как маркер из данной позиции может перемещаться в разные переходы в разной последовательности, то это позволяет описывать базовые операции алгебры событий, конъюнкцию, дизъюнкцию, последовательность[4].

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1.Королёва Д.А. Сравнительный анализ систем обработки сложных событий с открытым исходным кодом / Д.А. Королёва, Е. А. Данилина, С.В. Шибанов //Информационные технологии в науке и образовании. Проблемы и перспективы : сб. науч. ст. Всерос. межвуз. науч.практ. конф. (г. Пенза, 14 марта 2018 г.) / под ред. Л. Р. Фионовой. – Пенза : Изд-во ПГУ, 2018. – с.163 - 165.

2. Данилина Е.А. Системы обработки сложных событий или СЕР системы Е. А. Данилина,Д.А. Королёва, С.В. Шибанов // Информационные технологии в науке и образовании. Проблемы и перспективы: сб. науч. ст. Всерос. межвуз. науч.практ. конф. (г. Пенза, 14 марта 2018 г.) / под ред. Л. Р. Фионовой. – Пенза :Изд-воПГУ, 2018. – с.149 - 152.

3. W.M.P. van der Aalst and B.F. van Dongen. Discovering Petri Nets From Event Logs// K. Jensen et al.(Eds.): ToPNoC VII, LNCS 7480. – 2013. – P.372-422

4. Adaikkalavan R. Event Operators: Formalization, Algorithms, and Implementation Using Interval-Based Semantics. / R.Adaikkalavan. – Arlington.: MS Thesis, 2002. – p.4 – 5.