

УДК 004:042

ОБЗОР МОДЕЛЕЙ ПРЕДСТАВЛЕНИЯ АКТИВНЫХ ПРАВИЛ В ДИСКРЕТНО-СОБЫТИЙНЫХ ИНФОРМАЦИОННО-УПРАВЛЯЮЩИХ СИСТЕМАХ

С. В. Шибанов¹, Я. С. Шлепнев²

^{1,2}Пензенский государственный университет, Пенза, Россия

¹ sergey.v.shibanov@yandex.ru

² yaroslav.shlepnev@yandex.ru

Аннотация. Проведен обзор формальных и графических моделей для описания событийных процессов и выполнения активных правил в дискретно-событийных реагирующих информационно-управляющих системах. Рассматриваются различные подходы к представлению событий (включая использование условных и временных операторов, которые позволяют более точно управлять активацией и последовательностью выполнения правил). Особое внимание уделено операциям над событиями с учетом временной семантики их возникновения, что важно для правильной интерпретации сложных событий и их комбинаций. Также обсуждаются методы повышения эффективности алгоритмов обнаружения и устранения проблемных ситуаций при исполнении активных правил, таких как конфликты, заикливания и другие ошибки, возникающие при параллельной или последовательной активации событий.

Ключевые слова: активные правила, формальные модели, информационно-управляющие системы, графы отображения событий, сети Петри, алгебра логики

Для цитирования: Шибанов С. В., Шлепнев Я. С. Обзор моделей представления активных правил в дискретно-событийных информационно-управляющих системах // Вестник Пензенского государственного университета. 2024. № 4. С. 143–148.

Введение

Современные информационно-управляющие системы (ИУС) играют ключевую роль в различных отраслях промышленности, транспорта, энергетики и других секторах экономики. Эффективное управление этими системами требует оперативной обработки и анализа большого объема данных, поступающих в реальном времени. Важным аспектом является разработка алгоритмов, которые способны анализировать потоки событий и на их основе формировать управляющие воздействия, обеспечивая тем самым высокую адаптивность и устойчивость ИУС.

Правила типа «событие – условие – действие» (ЕСА) [1] обладают достаточно высокой выразительностью для описания сложных событий и реакций на них. Поэтому этот событийно-управляемый формализм широко используется для спецификации сложных информационно-управляющих систем [2], например, для управления в промышленных масштабах и повышения эффективности, особенно при интеграции с технологиями, такими как встроенные системы и сети датчиков.

Правила ЕСА используются для задания реакции системы на события и записываются в формате: «при наступлении набора событий, если выполняются определенные условия, выпол-

нить данные действия». Однако для систем с большим количеством компонентов и сложным поведением может быть трудно корректно задать такие правила [3].

Классический подход к описанию активных правил – это использование нотации «событие – условие – действие» (Event – Condition – Action, ECA) [4]. В этом подходе выделяются типы событий и их экземпляры. Все события делятся на группы по типам, каждому из которых присваиваются имя и параметры с указанными типами данных, выступающие в качестве входных данных для активного правила. Конкретное событие определенного типа, которое происходит в системе, называют экземпляром события; оно включает информацию о типе, времени возникновения и контексте, представляющем собой набор значений параметров.

Условие – это логическое выражение, основанное на контексте события, определяющее, должна ли система реагировать на событие или проигнорировать его.

Действие, прописанное в активном правиле, представляет собой цепочку управляющих операций, выполняемых как отклик на событие и направленных на управляемый объект ИУС. Параметры этих операций становятся выходными параметрами правила [5].

Проблемы исполнения активных правил

Встраивание активных правил (ECA) в дискретно-событийные реагирующие динамические системы реального времени направлено на оптимизацию процесса управления с помощью четко определенного набора правил. Активные правила применяются в информационно-управляющих системах реального времени [6], системах поиска информации и интеллектуального анализа текста, а также в сценариях мониторинга статистических изменений в веб-приложениях [7]. Тем не менее, в ходе выполнения активных правил возможны конфликты, связанные с их взаимодействием.

Конфликты активных правил считаются основной угрозой в процессе обнаружения и обработки событий.

Так, одной из проблемных ситуаций активных правил является несогласованность правил: два правила считаются несогласованными, если порядок их выполнения может привести к разным результатам; т.е. выполнение e_1 , а затем e_2 приводит к иному результату, чем выполнение e_2 , а затем e_1 . Смысловая несогласованность действий, выполняющихся в результате активации правил, может привести рассматриваемые объекты управления к некорректному состоянию и нарушить целостность информационно-управляющей системы.

Действие активного правила может являться событием другого правила, что приводит к последовательной активации целого набора правил. Без наличия специальных алгоритмов контроля этот процесс может продолжаться бесконечно, что приводит к проблеме незавершенности правил или ситуации закликивания [8].

В процессе взаимодействия активных правил может возникать состояние гонки (race condition) – ситуация, когда последовательная активация различных правил приводит к изменению состояния одного и того же объекта управления, при этом активация второго правила отменяет результат активации первого правила.

Одной из главных особенностей дискретных реагирующих систем являются наличие быстрой реакции на возникновения событий и их обработка в режиме реального времени. Тем не менее, возникновение описанных проблемных ситуаций может приводить к деградации пропускной способности и к нарушению целостности системы (дискретные системы в тот или иной момент времени должны находиться в определенном состоянии). В статье проводится обзор формальных и графических моделей и приводится набор операций над событиями с учетом временной семантики их возникновения с целью дальнейшего построения эффективных алгоритмов выявления и устранения проблемных ситуаций исполнения активных правил.

Обзор формальных моделей представления активных правил

Данный раздел посвящен проблеме представления и формализации правил, а также приводятся решения и попытки их реализации.

В настоящее время было предпринято множество попыток представить правила как в формальной, так и в графической форме. Формальное представление можно найти в языках запросов, таких как в реляционных и XML-базах данных [9].

В работе [5] для моделирования процесса взаимодействия активных правил используют нотацию взаимодействующих последовательных процессов (CSP, Communicating Sequential Processes), разработанную Ч. Хоаром. Взаимодействующие последовательные процессы (CSP) – формальный язык для описания моделей взаимодействия в параллельных системах, который относится к математическим теориям параллелизма, известным как исчисление процессов (или алгебра процессов).

Авторы проекта HiPAC [9], изучающие активные базы данных, разработали формальную модель, которая считается эталоном вербального представления правил для различных систем, расширяющую базовую нотацию ECA с добавлением времени исполнения, режима исполнения и пользовательских атрибутов правила.

Данные формальные представления применяются в различных информационно-управляющих системах, но имеют существенные недостатки относительно графических, не только с точки зрения полноты описания правил, но и с точки зрения конструирования правил и их взаимодействия. Описанные формальные модели учитывают критические аспекты взаимодействия правил, в ходе которых могут происходить различные проблемные ситуации (коллизии), но не позволяют конструировать алгоритмы, позволяющие выявить и устранить потенциальные угрозы процесса исполнения.

Обзор систем и моделей графического представления активных правил

Объектно-ориентированные базы данных (OODB) с подходом отображения, основанным на применении правил, представляют собой системы, в которых правила организуются и визуализируются в виде слоев. Это позволяет структурировать правила, улучшить их наглядность и управляемость, особенно в контексте сложных данных. Такой подход используется в различных моделях, например, **SAMOS-PN** в системе **SAMOS**, а также в моделях **Snoop** и **SnoopIB**, где графическое представление слоя правил способствует лучшему пониманию логики и структуры данных [10, 11].

Система **SnoopIB** (Snoop Interval-Based) позволяет моделировать процесс исполнения активных правил с использованием графа обнаружения событий (Event Detection Graph, EDG), в котором простые события отображаются в виде листовых узлов, а внутренние узлы представляют составные события. Система **SnoopIP** является развитием системы **Snoop**, которая, в свою очередь, основывается на **Sentinel**. Одной из особенностей данной системы является использованием операторов булевой алгебры для представления различных классов событий с учетом их временной семантики возникновения.

Так, оператор ЛЮБОЙ может представлять операцию конъюнкции, оператор АПЕРИОД позволяет представить события, которые могут происходить более одного раза в течение заданного интервала времени, а оператор ПЕРИОД работает аналогично АПЕРИОД, но с фиксированным интервалом активации события в пределах заданного интервала времени.

Тем не менее, граф обнаружения событий обладает следующими существенными недостатками:

- возможное создание дубликатов одних и тех же событий, что приводит к избыточности и увеличению объема памяти;

- недостаточно гибкие механизмы для учета условий активации правил, что ограничивает сложные сценарии исполнения;
- меньшая эффективность в обработке комбинаций событий по сравнению с более специализированными системами;
- отсутствие возможности выявления и устранения критических ситуаций исполнения.

Все это привело к необходимости возникновения более гибкой и адаптивной модели отображения.

Система **SAMOS** (Swiss Active Mechanism-based Object-oriented database System) – это объектно-ориентированная база данных с применением механизма активных правил. Эта система реализует собственную модель графического отображения активных правил, которая называется SAMOS-PN. Данная система основывается на классических сетях Петри, но имеет критический недостаток в виде отсутствия условных переходов для управления сложными процессами управления.

CCPN (Conditional Colored Petri Nets) – это разновидность цветных сетей Петри (Colored Petri Nets, CPN), которая добавляет условные переходы для управления сложными процессами. В классической цветной сети Петри каждый маркер (токен) может иметь цвет или атрибут, который помогает различать данные, проходящие через сеть. Однако в условной цветной сети Петри добавляется еще один уровень контроля – условия, которые позволяют гибко управлять переходами. Эти условия могут зависеть от значений токенов или от внешних параметров, что позволяет активировать или блокировать переходы в зависимости от определенных состояний.

По сравнению с графами обнаружения события и классическими сетями Петри условно раскрашенные сети Петри имеют следующие преимущества:

- отсутствие дублирования событий и контекстов событий;
- учитывается условие активации при обработке правил, что обеспечивает более точное управление активными правилами и адаптацию к различным контекстам и сценариям исполнения;
- обеспечиваются более эффективное обнаружение и обработка составных (сложных) событий, что делает его подходящим для сложных сценариев, где события могут взаимодействовать друг с другом.

Сравнительный анализ систем и моделей представления активных правил и наборов операторов алгебры логики

Описанные программные средства представляют активные правила различными способами. Системы Sentinel, Snoop, SnoopIB в качестве модели отображения используют графы обнаружения событий, в то время как система SAMOS основывается на применении классической сети Петри.

Для моделирования составных событий в указанных системах используются логические операции алгебры логики. Активные правила могут рассматриваться как предикаты, поскольку они определяют условия (предикаты), которые должны быть истинными для активации действия. Активные правила могут быть представлены в рамках исчисления предикатов, где условия и действия могут быть представлены как логические выражения. Например, активное правило может быть описано в виде формулы, включающей логические операторы, такие как AND, OR и NOT, и кванторы, такие как «для всех» (\forall) или «существует» (\exists).

С точки зрения операторов алгебры логики можно выделить следующие основные операции над событиями:

- конъюнкция событий (И): $I(e_1, e_2, \dots, e_n)$;
- дизъюнкция событий (ИЛИ): $ИЛИ(e_1, e_2, \dots, e)$;
- отрицание события (НЕ) (по интервалу): $НЕ(e_1)$ в интервале $[t_1, t_2]$;

- последовательность событий (ПОСЛЕД): ПОСЛЕД (e_1, e_2, \dots, e_n) ;
- любое событие (ЛЮБОЕ): ЛЮБОЕ $(m, e_1, e_2, \dots, e_n)$, где $m \leq n$, m – количество активированных событий, n – количество всех событий;
- замыкание события (ПЕРВ): ПЕРВ (e_1) в интервале $[t_1, t_2]$, т.е. первое появление события e_1 .

Описанные средства представления активных правил в той или иной степени реализуют основные логические операции над событиями. Тем не менее, с целью построения логики исполнения активных правил с учетом критических аспектов их взаимодействия и расширенных классов событий авторы рассмотренных систем вводят новые логические операции для учета временной семантики возникновения событий.

Система Snoop, основанная на графах отображения событий, расширяет набор логических операторов для представления периодических (с фиксированным интервалом возникновения) и аperiodических (без фиксированного интервала возникновения) событий.

Система SnoopIB предоставляет дополнительные операторы для поддержки обнаружения составных (сложных) событий, такие как операторы AND, NOT и PLUS, которые не были представлены в Snoop. Такое дополнение графа отображения событий является большим преимуществом по сравнению с моделью раскрашенной сети Петри (S-PN), применяемой в систему SAMOS. Тем не менее, наличие дубликатов в графах обнаружения событий все еще является существенным недостатком по сравнению с моделью CCPN, в которой отсутствует дублирование событий и состояний.

Классические сети Петри, применяемые в SAMOS (S-PN), используются для представления потоков входящих событий их параметров, в которой дочерние выражения представлены отдельно, что может привести к их дублированию в модели (является также большим недостатком системы и раскрашенных сетей Петри).

Модель условно раскрашенной сети Петри, в свою очередь, поддерживает все рассмотренные логические операторы, расширяя набор операторов логической функцией ОДНОВРЕМЕННО, которая отслеживает возникновение набора событий в течение заданного промежутка времени.

В табл. 1 представлен сравнительный анализ наличия логических операторов в описанных системах.

Таблица 1

**Сравнительный анализ наличия логических операторов
в системах и моделях представления активных правил**

Оператор алгебры логики	Sentinel (EDG)	Snoop (EDG)	SnoopIB (EDG)	SAMOS (PN)	CCPN
И	✓	–	✓	✓	✓
ИЛИ	✓	✓	✓	✓	✓
ЛЮБОЕ	✓	✓	✓	–	✓
ПЕРВ	–			✓	✓
НЕ	–	–	✓	✓	✓
ПОСЛЕД	–	✓	✓	✓	✓
ПЕРИОД	–	✓	✓	–	✓
АПЕРИОД	–	✓	✓	–	✓
ОДНОВРЕМЕННО	–	–	–	–	✓

Так, набор всех операторов с учетом временной семантики возникновения событий для описания сложных событий с использованием условно раскрашенной сети Петри может выглядеть следующим образом:

- конъюнкция событий (И): И (e_1, e_2, \dots, e_n) ;
- дизъюнкция событий (ИЛИ): ИЛИ (e_1, e_2, \dots, e) ;

- отрицание события (НЕ) (по интервалу): НЕ (e_1) в интервале $[t_1, t_2]$;
- последовательность событий (ПОСЛЕД): ПОСЛЕД (e_1, e_2, \dots, e_n);
- любое событие (ЛЮБОЕ): ЛЮБОЕ (m, e_1, e_2, \dots, e_n), где $m \leq n$, m – количество активированных событий, n – количество всех событий;
- замыкание события (ПЕРВ): ПЕРВ (e_1) в $[t_1, t_2]$, т.е. первое появление события e_1 ;
- периодическое возникновение: ПЕРИОД ($e_1, [t_1, t_2], t_3$), где $[t_1, t_2]$ – временной интервал, t_3 – частота возникновения события;
- аperiodическое возникновение: АПЕРИОД ($e_1, [t_1, t_2]$), где $[t_1, t_2]$ – временной интервал;
- одновременное возникновение событий: ОДНОВРЕМЕННО ($e_1, e_2, \dots, e_n, [t_1, t_2]$), где $[t_1, t_2]$ – временной интервал.

Заключение

Таким образом, предложенное формализованное и математически обоснованное представление событий в виде условной раскрашенной сети Петри с применением логических операторов, учитывающих временную семантику возникновения событий, позволяет выстраивать эффективные алгоритмы выявления и устранения коллизий исполнения активных правил, что приводит к повышению эффективности управления дискретно-событийных информационно-управляющих систем.

Список литературы

1. McCarthy D., Dayal U. The architecture of an active database management system // ACM SIGMOD Record. 1989. № 18 (2). P. 215–224.
2. Abadi D. J., Carney D., Çetintemel U. [et al.]. Aurora: a new model and architecture for data stream management // The VLDB Journal. 2003. № 12 (2). P. 120–139.
3. Augusto J. C., Nugent C. D. A new architecture for smart homes based on ADB and temporal reasoning // Toward a Human-Friendly Assistive Environment. 2004. Vol. 14. P. 106–113.
4. Caroprese L., Truszczyński M. Declarative Semantics for Active Integrity Constraints // Logic Programming. Lecture Notes in Computer Science. 2008. № 5366. P. 269–283.
5. Шибанов С. В., Скоробогатько А. А., Лысенко Э. В. Интегрированная модель активных правил // Математическое и программное обеспечение систем в промышленной и социальной сферах. 2011. № 1-1. С. 41–46.
6. Шибанов С. В., Шлепнев Я. С. Сервис потоковой обработки событий и исполнения активных правил // Математическое моделирование и суперкомпьютерные технологии : тр. XXI Междунар. конф. (г. Н. Новгород, 22–26 ноября 2021 г.). Н. Новгород : Национальный исслед. Нижегород. гос. ун-т им. Н. И. Лобачевского, 2021. С. 403–407.
7. Zimmer D., Meckenstock A., Unland R. Using Petri Nets for Rule Termination Analysis // Proceedings of the workshop on on Databases: active and real-time. 1997. P. 29–32. URL: researchgate.net>
8. Couchot A. Improving Termination Analysis of Active Rules with Priorities // DEXA. 2003. Vol. 3. P. 846–855.
9. Dayal U., Blaustein B., Buchmann A. P. [et al.]. The HiPAC Project: Combining Active Databases and Timing Constraints // SIGMOD Record. 1988. Vol. 17. P. 51–70.
10. Gatzu S., Dittrich K. R. SAMOS: An active object-oriented database system // IEEE Quartely Bulletin on Data Engineering. 1993. Vol. 15. P. 23–26.
11. Dittrich K. R., Fritsch H., Gatzu S. [et al.]. SAMOS in hindsight: experiences in building an active object-oriented DBMS // Information Systems. 2003. Vol. 28. P. 369–392.

Информация об авторах

Шибанов Сергей Владимирович, кандидат технических наук, доцент, доцент кафедры «Математическое обеспечение и применение электронных вычислительных машин», Пензенский государственный университет.

Шлепнев Ярослав Сергеевич, аспирант, Пензенский государственный университет.

Авторы заявляют об отсутствии конфликта интересов.