дения пользователей системы разного типа. Показано, что применение различных алгоритмов компоновки узлов графа на плоскости в качестве механизма взаимодействия пользователя при графическом представлении данных способствует выявлению групп пользователей, обладающих общими характеристиками, и может служить отправной точкой для изучения контактов пользователей СМДП. Дальнейшее направление исследований связано с разработкой графических способов представления динамики изменения структурных связей пользователя, поскольку многие виды электронного финансового мошенничества характеризуются большим числом подозрительных транзакций, совершаемых за короткий промежуток времени.

Работа выполняется при частичной финансовой поддержке, осуществляемой в рамках проектов Евросоюза и MASSIF.

СПИСОК ЛИТЕРАТУРЫ

Al-Khatib A. Electronic Payment Fraud Detection Techniques// World of Computer Science and Information Technology J. (WCSIT). 2012. Vol. 2. P. 137–141.

E. S. Novikova

VISUALIZATION OF THE LOGS OF THE MOBILE MONEY TRANSFER SERVICES FOR DETECTING FRAUDSTER ACTIVITY

This paper presents the first results of the research devoted to the design of the visualization techniques purposed to detect anomaly activity in mobile money transfer services. Mobile money services are currently being deployed in many markets across the world, they are widely used for domestic and international remittances. However these Mobile money services can be used for money laundering and other illegal financial operations due to rapidity of the electronic transactions.

Visual analytics, mobile money transfer, anomaly detection, interactive data visualization

УДК 66-933.6

Р. А. Нечитайленко

Построение алгоритма логической фильтрации событий в территориально распределенной системе управления на основе графов зависимостей

Рассматривается возможность применения метода логической фильтрации на основе графов зависимостей. Предлагается алгоритм логической фильтрации событий в территориально распределенной системе управления на основе графов зависимостей.

Системы обнаружения аномалий, графы зависимостей, логическая фильтрация информации, территориально распределенная информационная система

Существующие механизмы логической фильтрации событий в территориально распределенной системе управления основаны на установлении причинно-следственных отношений на множестве событий. Для этой цели в настоящее время используется несколько подходов, обладающих различной выразительностью и эффективностью. В табл. 1 приведена оценка преимуществ и недостатков существующих методов [1].

Из таблицы следует, что наибольшим числом преимуществ для решения задач логической фильтрации информации в территориально распределенной системе управления обладает метод на основе графов зависимостей. Данный метод и лег в основу алгоритма фильтрации событий.

Метод логической фильтрации на основе графов зависимостей. Подходом для решения

проблемы определения первопричины является построение графа зависимостей. Граф зависимости – это граф, показывающий причинно-следственные отношения на множестве событий [2].

В простейшем случае граф зависимостей будет являться двухсторонним графом: соответствующей первопричине будет соответствовать один или несколько признаков. При появлении набора признаков выявляется проблема обнаружения всех первопричин, которые смогут полностью объяснить все признаки. Если же признак связан только с одной первопричиной, как на рис. 1, проблема диагноза может быть просто решена нахождением первопричины, которая генерирует этот признак.

Однако при формировании графа признак может быть связан более чем с одной первопричиной, поскольку первопричина может быть сгенерирована более чем одним признаком, и наоборот, признак может быть сгенерирован более чем одной первопричиной. На рис. 2 показан универсальный случай, когда у графа зависимостей нет простых отношений между первопричинами и сгенерированными признаками. В этом случае задача состоит в определении набора первопричин, ответственных за сгенерированные признаки. Проблема заключается в том, что один признак может быть сгенерирован различными первопричинами, поэтому необходима дополнительная информация, чтобы из множества первопричин выбрать необходимую.

Таблица 1

Название метода	Преимущества	Недостатки
Метод логической фильтрации на основе правил	Простота и интуитивность представления знаний	Сложность выработки конкретного набора правил
Метод логической фильтрации на основе моделей	Адаптивность по отношению к изменению структуры сети. Модель диагностики сети формируется в результате композиции описаний поведения входящих в сеть элементов	Отсутствие строго теоретического базиса и, как следствие, невозможность проведения формальной верификации моделей. Достаточно высокие вычислительные расходы, связанные с моделированием управляемой системы
Метод логической фильтрации на основе модели переходов	Возможность применения формальных методов верификации	Необходимость создания большого числа графов
Метод логической фильтрации на основе кодовых книжек	Принятие кодовых книжек, позволяющее уменьшить размер большой корреляционной матрицы. Снижение требований к вычислительным ресурсам и уменьшение времени принятия решений	Необходимость правильного составления исходной матрицы. Фактор сжатия корреляционной матрицы может быть как очень большим, так и очень малым, т. е. составление кодовых книжек может оказаться неэффективным по их размерам. Необходимость выбора при получении исходной матрицы нескольких книжек. Отсутствие гарантии исключения ошибок для простейшей книжки
Метод логической фильтрации на основе базы знаний о неисправностях	Адаптивность к изменению решения	Сложность разработки алгоритма выборки релевантного случая
Метод логической фильтрации на основе графов зависимостей	Хорошая теоретическая обоснованность метода. Возможность формальной верификации модели. Корректность сводится к отсутствию циклов в графе зависимостей событий. Возможность компактного представления графа в виде булевых функций	Отсутствие адаптивности к изменениям структуры объекта управления. (Недостаток может быть устранен за счет комбинации с модельным подходом.)

На рис. 2 первопричина C_1 вызывает признак S_1; первопричина C_2 вызывает признаки S_1, S_2,.S_4; первопричина C_3 вызывает признаки S_2, S_3; первопричина C_4 вызывает признаки S_3, S_4. Таким образом, возникновение любого признака из набора $\{S_1,\{ S_1, S_2, S_4 \},\{ S_2, S_3\},\{S_3, S_4\}\}$ может указывать точно на одну из первопричин $\{C\}$.

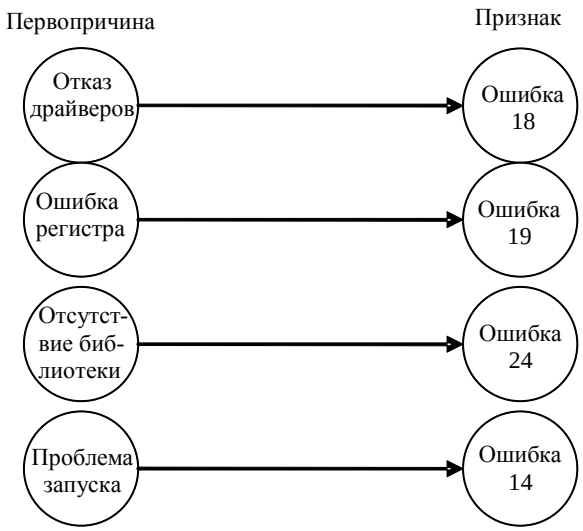

Рис. 1

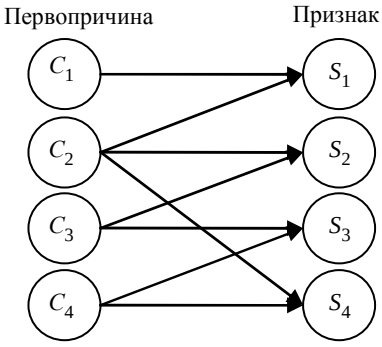

Рис. 2

В действительности же комбинация различных признаков, возникающих вместе, может просматриваться как выведенный признак, соответствующий одной из первопричин. Если признаки не теряются, они все должны быть отображены в одной из групп признаков.

График зависимости может также быть недвусторонним, если первопричина непосредственно генерирует признак, что может вызвать возникновение некоторого другого события в сети, которое приводит к генерации признака. Другими словами, у графов зависимостей, соединяющих первопричины и признаки, могут быть посреднические узлы, которые являются вторичными или полученными признаками.

Алгоритм логической фильтрации событий на основе графов зависимостей. Событием (event), с точки зрения мониторинга, считаем всякое изменение состояния системы, происходящее в некоторый момент времени время примем в единичной дискретной системе). Будем считать, что события происходят мгновенно, т. е. можно точно указать момент времени, в который объектом было сгенерировано данное событие. Среди потока событий, поступающего от элемента информационной инфраструктуры, принято выделять события, соответствующие ненормальному состоянию системы (fault), т. е. такому состоянию, при котором реальное поведение системы отличается от ожидаемого. Объекты модели, отражая поведение реальных элементов распределенной системы, взаимодействуют между собой, что позволяет классифицировать множество событий по *критерию причинности* на непосредственные и косвенные. Причиной непосредственных событий является изменение состояния моделируемого объекта фрагмента системы. Примером может служить падение/превышение коэффициента использования интерфейса, рестарт сервера баз данных вследствие перепада напряжения, превышение уровня ошибок в кадрах вследствие увеличения уровня наводок. Иными словами, генерация события происходит за счет воздействия среды на моделируемый объектом фрагмент системы. Косвенные события генерируются объектом вследствие его взаимодействия с другими объектами модели. Причиной отказа объекта в этом случае является отказ другого объекта, связанного с данным некоторой ассоциацией. Например, отказ интерфейса, очевидно, приведет к отказу LAN-точки, отказ которой в свою очередь вызовет отказ IP-точки. Связи между событиями классифицируют на причинные (causal) и временные (temporal) [3]. Причинная связь, как следует из названия, отражает причинно-следственную связь между событиями. Временная связь налагает помимо этого ограничения на моменты возникновения связанных событий. Сбои принято классифицировать на жесткие (hard) и мягкие (soft). Жесткие сбои ассоциируются с полной потерей системой своих функций вследствие ее выхода из строя или выхода из строя используемых элементов системы. Соответственно, мягкие сбои характеризуются неполной потерей работоспособности системы [4]. Введем несколько базовых определений и понятий. Будем говорить, что событие e коррелирует с множеством событий $e_1, e_2, \ldots, e_k$, если $e_1, e_2, \ldots, e_k$, входя в отношение друг с другом и с событием e, определяют образец поведения системы при появлении события e. Будем обозначать этот факт $e \Rightarrow \{e_1, e_2, ..., e_k\}$.

Определение. Структура событий (causal event structure) задается парой $\Im = \langle E, \to \rangle$, где E – множество событий; $\to \subseteq E \times E$ – причинно-следственное бинарное отношение, задающее слабый частичный порядок на множестве событий, т. е. данное отношение обладает следующими свойствами. Для всяких событий $e_a, e_b, e_c \in E$:

$$- \frac{e_a \to e_b \ , \ e_b \to e_c}{e_a \to e_c} \qquad - \textit{транзитивность}$$

и

$$- \ e_a \to e_a \qquad - \textit{антисимметричность}.$$

Интуитивный смысл отношения $e_1 \to e_2$: событие e_1 является причиной e_2 (возможно, причиной нескольких событий). Транзитивное замыкание отношения $\to$ обозначим $\prec$, именно

$$e_a \prec e_b \Leftrightarrow \exists e_1, e_2 ... e_n \in E : e_a \to e_1 \to e_2 \to ... \to e_n \to e_b.$$

Наглядной формой представления структуры событий является событийный граф. Основным свойством данного графа является ацикличность, которая следует из транзитивности и антисимметричности структуры событий. В вершинах данного графа располагаются события, а ребра выражают причинно-следственное отношение. В качестве примера рассмотрим граф на рис. 3, из которого следует, что $8 \Rightarrow \{2, 3, 6\}$, так как появление события 8 приведет к появлению событий 2, 3, 6. Также можно заключить, что $8 \Rightarrow \{1, 2, 3, 6\}$ вследствие транзитивности отношения причинности, так как появление события 2 приведет к появлению события 1. С другой стороны, появление событий {3, 6} еще не означает наличия события 8, так как для появления события 8 необходимо также наличие события 2. По транзитивности отношений же событию $11 \Rightarrow \{4, 5, 10\}$ – события 4, 5, 10.

Определение. Пусть задана структура $\upsilon = \langle E, \to \rangle$. *Шаговым эффектом события* $e \in E$ StepEff(e) назовем множество событий $S = \text{StepEff}(e) = \{e' | e \to e'\}$.

Расширим отношение $\Rightarrow$ до множеств, а именно $S \Rightarrow S'$, если

$$\forall e \in S \ \exists e' \in S' : e \prec e',$$
$$!\exists e \in S \ \forall e' \in S' : e' \prec e.$$

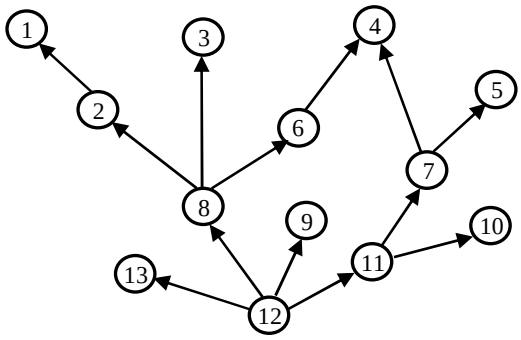

Рис. 3

Для формализации семантики оператора $\Rightarrow$ вводятся следующие правила вывода :

(Effects)

$$\{e\} \Rightarrow \text{StepEff}(e) \ \text{ if } \ \text{StepEff}(e) \neq \varnothing$$

(Reflexivity) $S \Rightarrow S$

(Transitivity) $$\frac{S_1 \Rightarrow S_2 \quad S_2 \Rightarrow S_3}{S_1 \Rightarrow S_3}$$

(Union) $$\frac{S_1 \Rightarrow T_1 \quad S_2 \Rightarrow T_2}{S_1 \cup S_2 \Rightarrow T_1 \cup T_2}$$

В рамках описанной семантики система правил вывода полна и непротиворечива.

Например, можно показать что, $8 \Rightarrow \{1, 2, 3, 6\}$ исходя из $8 \Rightarrow \{2, 3, 6\}$, т. е.

1. $\{2\} \Rightarrow \{1\}$ (Effect) ;
2. $\{3, 6\} \Rightarrow \{3, 6\}$ (Reflexivity) ;
3. $\{2, 3, 6\} \Rightarrow \{1, 3, 6\}$ (Union) ;
4. $8 \Rightarrow \{2, 3, 6\}$, $\{2, 3, 6\} \Rightarrow \{1, 3, 6\}$, следовательно, $8 \Rightarrow \{1, 2, 3, 6\}$ (Transitivity).

Следует заметить, что оператор $\Rightarrow$ не является простым расширением оператора $\to$ до множеств. В частности, согласно системе правил вывода оператор $\Rightarrow$ рефлексивен, в то время как $\to$ иррефлексивен.

Определение. Пусть задана структура $\upsilon = \langle E, \to \rangle$. Рангом события *e* назовем величину

$$\text{Rank}(e) = \text{Max}\left(\{\text{Rank}(e') \,|\, e' \in \text{StepEff}(e)\}\right) + 1.$$

Наглядно, что ранг элемента равен высоте максимального подэлемента +1.

Определение. Ранг множества событий $S \subseteq E$ равен

$$\text{Rank}(S) = \text{Max}\left(\{\text{Rank}(e) \,|\, e \in S\}\right).$$

Теорема. Пусть задана структура $\Im = \langle E, \to \rangle$ и $S \subseteq E$ – подмножество априорных событий, полученных из внешней среды. Тогда минимальное множество причин событий S MinCause (S) должно обладать следующими свойствами:

1) *корректностью*: $\mathrm{MinCause}(S) \Rightarrow S$;

2) *оптимальностью*: для любого $S' : S' \Rightarrow S$ имеем $\mathrm{Rank}(S') \le \mathrm{Rank}(\mathrm{MinCause}(S))$.

Метод логической фильтрации на основе графов зависимостей, таких, как возможность формальной верификации, возможность компактного представления графа в виде булевых функций, а также данный метод, имеет хорошую теоретическую обоснованность. Недостатком данного метода является отсутствие адаптивности к изменениям структуры, но данный недостаток может быть устранен комбинированием данного метода с модельным подходом. Таким образом, метод логической фильтрации на основе графов зависимостей наиболее предпочтителен для разработки подсистемы логической фильтрации событий.

СПИСОК ЛИТЕРАТУРЫ

1. Dinesh Chandra Verma. Principles of Computer Systems and Network Management, Springer Science. New York: Business Media, 2009.

2. Masum Hasan, Binay Sugla, Ramesh Viswanathan. A Conceptual Framework for Network Managment Event Correlation and Filtering Systems //Proc. of the sixth IFIP/IEEE Intern. Symposium, Boston, 1999. MA. P. 5–9.

3. Rozemeijer E., Van Bon J., Verheijen T. Frameworks for IT Management: A Pocket Guide. Zaltbommel: Van Haren Publishing, 2007.

4. Kenneth R. Sheers. HP OpenView Event Correlation Services // Hewlett-Packard J. 1996. Oct. P. 6–7.

R. A. Nechitaylenko

DEMENSIONALITY REDUCTIONS METHODS FOR FILRETING INFORMATION IN GEOGRAPHICALLY DISTRIBUTED INFORMATION SYSTEM

The possibility of applying the method of logical filtering based on dependency graphs. We propose an algorithm logic filter events in geographically distributed control system based on dependency graphs.

Anomaly detection system, dependency graphs, logical filtering information, geographically distributed information system

УДК: 20.53.19, 28.23.13

И. В. Петухов

Распределенное выполнение алгоритмов построения деревьев решений с использованием библиотеки для анализа данных и концепции Map-Reduce

Описывается вариант распараллеливания алгоритмов построения деревьев решений с использованием библиотеки анализа данных на основе блочной структуры. Кроме того описывается способ взаимосвязи библиотеки и вычислительного кластера, основанного на технологии Map-Reduce. Описываются результаты экспериментального запуска алгоритма в разных средах.

Деревья решений, параллельные алгоритмы, распределенные вычисления

Современную жизнь невозможно представить без алгоритмов интеллектуального анализа данных, которые используются даже в обычных магазинах. Однако применение данных алгоритмов требует, как правило, большой вычислительной мощности в связи с тем, что для выявления закономерностей нужны большие объемы информации. Такие мощности не может предоставить локальная нераспределенная система. Даже с учетом темпов научно-технического прогресса такие задачи не решаются одним компьютером в одиночку. Для обработки огромного количества данных требуется параллельное выполнение алгоритмов data mining.