

УДК 004.891.3

С. Ф. ЧАЛИЙ, І. Б. ПРИБИЛЬНОВА

## ПОБУДОВА СИТУАЦІЙНОГО ПРЕДСТАВЛЕННЯ ЗНАТЬ НА ОСНОВІ АНАЛІЗУ ЛОГІВ

В роботі розглянуто проблему побудови ситуаційних моделей представлення знань з використанням логів процесів. Розроблено методи пошуку патернів ситуації у вигляді послідовностей подій на трасах логів, а також побудови схеми дій ситуації, що використовує отримані патерни. Метод пошуку ситуацій комбінує використання обмежень по охопленню патерном трасам логів і по частоті появи послідовності подій ситуації у файлі логів. Метод побудови схеми дій ситуації призначений для побудови workflow – схеми ситуації з використанням алгоритму process mining на основі патерну у вигляді послідовності ситуацій, що дає можливість побудувати модель процесу у вигляді послідовності ситуацій із визначенням обмежень для кожної ситуації. Використання запропонованих методів дозволяє адаптувати модель ситуації та процесу в цілому до поточного стану предметної області з використанням відповідних правил.

**Ключові слова:** представлення знань, інтелектуальний аналіз процесів, ситуація, ситуаційне обчислення, обмеження.

В работе рассмотрена проблема построения ситуационных моделей представления знаний с использованием логов процессов. Разработаны методы поиска паттернов ситуации в виде последовательностей событий на трассах лога, а также построения схемы действий ситуации с использованием полученных паттернов. Метод поиска ситуаций комбинирует использование ограничений по охваченным патерном трассам лога и по частоте появления последовательности событий ситуации в файле лога. Метод построения схемы действий ситуации предназначен для построения workflow - схемы ситуации с использованием алгоритма process mining на основе паттерна в виде последовательности ситуаций, что дает возможность построить модель процесса в виде последовательности ситуаций с определением ограничений для каждой ситуации. Использование предложенных методов позволяет адаптировать модель ситуации и процесса в целом к текущему состоянию предметной области с использованием соответствующих правил.

**Ключевые слова:** представления знаний, интеллектуальный анализ процессов, ситуация, ситуационное вычисления, ограничения.

The problem of constructing situational models of knowledge representation using logs of processes is considered in the paper. The methods of searching for situation patterns in the form of sequences of events on log lines are developed, as well as the construction of a scheme of action of the situation using the obtained patterns. The situation search method combines the use of constraints on the paths covered by the log and the frequency of occurrence of the sequence of situation events in the log file. The method of constructing the scheme of the situation is designed to construct a workflow - a scheme of the situation using the process mining algorithm on the basis of the pattern in the form of a sequence of situations, which makes it possible to construct a model of the process in the form of a sequence of situations with the definition of constraints for each situation. Using the proposed methods allows us to adapt the model of the situation and the process as a whole to the current state of the subject using the appropriate rules.

**Keywords:** knowledge representation, processes mining, situation, situational calculation, constraints.

**Вступ.** Інтелектуальний аналіз процесів направлений на побудову їх моделей на основі дослідження записів про їх поведінку. Побудова ситуаційних моделей представлення знань в задачах інтелектуального аналізу процесів (process mining) передбачає пошук шаблонів поведінки дискретних процесів у конкретних ситуаціях, що визначаються станом предметної області. Такі патерни повинні відповідати заданій множині обмежень [1].

Поведінка дискретних процесів, як правило, фіксується в файлах логів у вигляді послідовності подій. Кожна з подій відображує виконання або зміну стану однієї з операцій процесу. Послідовний набір подій, що відповідає одній реалізації процесу, становить трасу логів. Побудова послідовностей дій процесу на основі аналізу його логів виконується методами process mining [2]. В загальному вигляді задача пошуку шаблонів поведінки полягає в визначенні послідовностей подій із трас логів з обмеженням на мінімальну кількість появ шаблону в файлі логів та подальшій побудові схеми подій одним з методів process mining. Однак використання лише одного обмеження у більшості випадків є недостатнім для пошуку патернів. Зависокий поріг дозволить отримати лише найбільш загальні ситуативні патерни виконання процесу, що не дозволить описати його поведінку у нетипових або виняткових ситуаціях, які виникають внаслідок непередбачених зовнішніх впливів. Результатом низького порогу є зavelика

кількість патернів, що не дозволяє узагальнити опис відповідної ситуації.

Тому вирішення проблеми побудови ситуаційних моделей на основі аналізу логів потребує врахування додаткових обмежень та відповідного переходу до інтелектуального аналізу процесів в обмеженнях, на основі поєднання методів process mining та constraint programming (програмування в обмеженнях). Програмування в обмеженнях становить собою парадигму, у відповідності до якої властивості рішення задаються у вигляді обмежень. Останні задають відносини між змінними. Рішенням задачі є кінцевий набір значень, що задовольняє всім обмеженням. В якості змінних, що характеризують ситуацію, в задачах інтелектуального аналізу процесів виступають атрибути подій. Атрибути характеризують дії процесу, об'єкти предметної області, з якими взаємодіє процес.

Зазначене свідчить про актуальність поєднання переваг обох підходів при побудові ситуаційних моделей представлення знань.

**Аналіз останніх досліджень і публікацій.**

Ситуаційне представлення знань відображує темпоральні знання у вигляді наборів ситуацій [3]. Стан кожної ситуації може бути описано засобами темпоральної логіки, а зміна станів задається як послідовність подій, що відображають результати виконання відповідних дій [4].

Однак прийнятий в ситуаційному обчисленні

підхід передбачає, що кожна ситуація відповідає результатам однієї дії процесу, тобто переходу між ситуаціями необхідно виконувати після кожної дії [4, 5]. Недоліком такого підходу є побудова «плаского», неструктурованого опису процесу в цілому, без виділення проміжних результатів.

Аналогічний неструктурований workflow - опис формується методами process mining в результаті аналізу логів процесів [2]. Така модель послідовності дій процесу задає схему його роботи в цілому, без визначення конкретних умов (ситуацій) для вибору відповідних послідовностей дій. Тобто отримана в результаті інтелектуального аналізу модель не дозволяє виявити причинно-наслідкові зв'язки для виконання дій та в подальшому адаптувати і виконувати процес на основі логічного виводу на таких знаннях.

Таким чином, питанням побудови заснованого на знаннях представлення дискретних процесів не приділяється достатньо уваги.

**Метою цієї статті** є побудова методу пошуку патернів ситуації у вигляді послідовностей подій на трасах логу, а також методу побудови схеми дій ситуації, що використовує отримані патерни. Це дає можливість на основі аналізу логу розділити процес на ситуації та представити знання про кожну ситуацію у вигляді алгоритму дій та обмежень на ці дії з урахуванням властивостей об'єктів, з якими оперує процес. Внаслідок такого рішення послідовність дій у конкретній ситуації вибирається з використанням правил вибору дій.

#### Методи побудови ситуаційних моделей представлення знань.

Ситуаційна модель представлення знань містить у собі логічний опис стану ситуації; набір послідовностей дій, що реалізують окремі підпроцеси, а також обмеження для ситуації.

Дана модель використовується в рамках ситуаційного підходу до подання і моделювання темпоральних знань. У відповідності до даного підходу, знання про дискретні процеси представлені у формі множини ситуацій. Для кожної ситуації задаються умови виникнення та послідовність дій, тобто опис алгоритму поведінки процесу у даній ситуації. Після завершення заданого алгоритму змінюється стан предметної області – тобто виникає нова ситуація. Пов'язані з ситуацією обмеження задають можливі послідовності дій процесу в залежності від поточного стану предметної області.

Послідовність ситуацій задає опис дискретного процесу. Перехід між ситуаціями відбувається на транзакційній основі, тобто лише після повного завершення пов'язаного із ситуацією алгоритму. Часткове виконання алгоритму свідчить про те, що поточна ситуація продовжується.

У відповідності до наведеного опису ситуації як елемента дискретного процесу, можна зробити висновок, що при записі поведінки процесу ситуація відображується у вигляді набору підмножин послідовностей подій, заданих на різних трасах логу. Обмеження на виконання дій процесу записані в файлі

логу у вигляді відповідних значень атрибутів подій. Останні задають значення атрибутів об'єктів, з якими взаємодіє процес під час виконання цих дій.

Загальна проблема пошуку ситуацій на основі аналізу логів зводиться до пошуку послідовностей подій, що зустрічаються у заданій кількості трас логу.

Послідовність подій  $E_i$ , що відповідає  $i$ -ситуації, визначається наступним чином:

$$E_i = \left\{ e_{ij} \mid \forall e_{ij} \exists e_{ij-1} : e_{ij-1} \rightarrow e_{ij}, E = \bigcup_i E_i \right\}, \quad (1)$$

де  $e_{ij}$  – подія з траси логу процесу, що відповідає дії в  $i$ -й ситуації;

$e_{ij-1}$  – подія, що є попередньою для  $e_{ij}$  на трасі логу;

$E$  – множина всіх подій логу.

Вирішення проблеми пошуку ситуації потребує пошуку пар множин (послідовності подій, траси логу) при обмеженнях на відносну кількість трас:

$$s_i = \left\{ (E_i, \Pi_k) \mid \frac{|\Pi_k|}{|\Pi|} \geq \theta, L = \bigcup_k \Pi_k, E_i \subseteq \Pi_k \right\}, \quad (2)$$

де  $L$  – лог процесу;

$s_i$  –  $i$ -ситуація;

$\Pi_k$  – підмножина трас логу, для яких відбувалась ситуація  $s_i$ ;

$\theta$  – поріг для виявлення патернів ситуації.

Для подальшої уніфікації розгляду ситуації та методів її побудови відображення множини подій будемо формувати у двійковому вигляді, тобто якщо елемент  $e_{ij}$  множини подій логу  $E$  належить до потрібної ситуації  $s_i$ , то його відображення  $e_{ij}^* = 1$ , в іншому випадку  $e_{ij}^* = 0$ . Аналогічно для трас – якщо на трасі  $\pi$  виникли події ситуації, то відображення для траси приймає значення 1:

$$\pi \in \Pi_k \Rightarrow \pi^* = 1, \quad (3)$$

де  $\pi$  – траса логу;

$\pi^*$  – двійкове відображення траси логу.

У відповідності до виразу (2), для опису ситуації як послідовності подій використовують два обмеження:

– по частоті появи ситуації, тобто  $\frac{|\{\Pi_k\}|}{|\Pi|} \geq \theta$ ;

– по охопленим трасам логу  $\Pi_k$ .

Перше обмеження з урахуванням двійкового представлення відображень подій та трас обчислюється через суму трас, для яких виконується умова (3):

$$\frac{\sum_l \pi_{kl}^*}{|\Pi|} \geq \theta, \quad (4)$$

де  $\pi_{kl}^*$  – двійкове відображення траси логу, на якій зафіксовано патерн ситуації.

Оскільки кількість трас у файлі логу однакова для всіх ситуацій на значних проміжках часу, то при пошуку ситуацій знаменником у виразі (4) можна знехтувати.

Друге обмеження дозволяє визначити підмножину трас, які є суттєвими для досягнення результатів процесу з точки зору користувача. Наприклад, при виконанні процесу з мінімальними витратами часу або інших ресурсів.

Поріг (4) може бути встановлений з використанням двох підходів. У відповідності до першого, потрібно апріорно задати кількість трас логу, на яких виникала потрібна ситуація. Згідно другого підходу, доцільно комбінувати обидва обмеження та шукати мінімальне значення  $\theta$ , при умові виконання обмеження по охопленню трасам.

Запропонований метод пошуку патернів ситуації як множини подій містить у собі наступні етапи.

Етап 1. Побудова матриці розміру  $|E| \times |\Pi|$  для відображення у двійковому вигляді подій на трасах логу. Елементи даної матриці мають значення «1» в тому випадку, якщо подія присутня на трасі логу, та «0» у протилежному.

Етап 2. Визначення події  $e_{ij}$  з мінімальним значенням  $\sum_l \pi_{kl}^*$ . При апріорному визначенні  $\theta$  виконується перевірка обмеження (4).

Етап 3. Видалення трас, в яких не представлена подія  $e_{ij}$ .

Етап 4. Перевірка обмеження по охопленню трас логу. У випадку невиконання даного обмеження подія  $e_{ij}$  видаляється, а траса повертається до ситуації. В протилежному –  $e_{ij}$  включається до складу ситуації.

Етап 5. У випадку виключення події виконується перевірка на цілісність ситуації у відповідності до умови (5).

$$\exists \pi_k : \forall e_{ij} \in E_i \exists e_{i-1} \rightarrow e_{ij}, e_{i-1}, e_{ij} \in \pi_k, \quad (5)$$

де  $\pi_k$  – траса логу, що задовольняє обмеженню по охопленню трас.

Якщо умова (5) виконується, то перейти до етапу 2. В іншому випадку виділяються дві ситуації. По кожній з нових ситуацій робота алгоритму починається з етапу 1.

Таким чином, метод комбінує використання обмежень по охопленню трасам логу і по частоті появи послідовності подій ситуації. Спочатку розглядається подія з мінімальним значенням  $\sum_l \pi_{kl}^*$

та робиться спроба включати дану подію до складу ситуації. При виконанні спроби видаляються траси, які не містять у собі необхідної події. Видалення трас зменшує охоплення для ситуації, що розглядається. Тому необхідно перевірити, чи задовільняється дане обмеження. Якщо обмеження не виконується, то поточну подію необхідно видалити із даної ситуації.

Потім розглянуті кроки повторюються для іншої події. Додатково перевіряється обмеження на цілісність ситуації, тобто для кожної пари подій  $(e_{i-1}, e_{ij})$  ситуації повинна існувати хоча б одна траса, на якій вони записані послідовно. Результатом даного методу є послідовність подій, що складає ситуацію.

Результуюча послідовність залежить в першу чергу від визначення обмеження охоплення. В загальному випадку траси можуть бути віднесені до множини  $\Pi_k$  за різними критеріям, представленими у вигляді атрибутів подій.

Семантика цих критеріїв залежить від виду процесу. Наприклад, для бізнес-процесів у якості критеріїв можуть бути вибрані: ролі та прізвища виконавців, підприємство або країна, де виконувався процес. Для процесів у соціальних мережах критеріями можуть виступати групи персональних даних користувачів, проміжки часу, коли користувачі взаємодіяли в мережі, тощо. Для процесів розробки програмного забезпечення – категорія програмного продукту.

Отримана в результаті застосування розробленого методу послідовність подій ситуації може містити різні значення частоти появи для різних подій. Це означає, що на різних трасах логу виконання дій процесу в конкретній ситуації йшло в різній послідовності, в залежності від поточного стану об'єктів, з якими оперує процес. Тому для отримання представлення знань про дії процесу необхідно інтегрувати результати пошуку у парадигму process mining.

Ідея такої інтеграції полягає у фільтрації складових трас логу та видаленні подій, що не мають відношення до отриманої ситуації.

Метод побудови схеми дій ситуації містить у собі розглянутий вище пошук патернів, фільтрацію подій та побудову workflow – схеми з використанням алгоритму process mining. Метод складається з наступних етапів.

Етап 1. Формування обмежень по охопленню трас. Вхідними даними етапу є множина атрибутів об'єктів, з якими взаємодіє процес, а також значень цих атрибутів. Результатом етапу є множина  $\Pi_k$ .

Етап 2. Побудова патернів ситуацій у вигляді послідовностей подій у відповідності до розглянутого вище методу.

Етап 3. Фільтрація підмножини  $\Pi_k$  трас логу. Зі вказаної підмножини трас видаляються події, що не належать до знайденого патерну. Результатом даного етапу є відфільтрований лог, який може бути використаний для побудови схеми дій процесу.

Етап 4. Побудова workflow – схеми для знайденої ситуації з використанням алгоритмів process mining.

Етап 5. Доповнення дій із workflow – схеми обмеженнями на виконання у вигляді пар «атрибут-значення». На даному етапі використовуються вхідні дані – обмеження охоплення трас, а також події з патерну ситуації.

Кожна подія у складі логу містить у собі опис значень атрибутів відповідних об'єктів. До цих

атрибутів належать характеристики дії, яка зафіксована у вигляді події логу. Це дає можливість співставити дії ситуації та значення відповідних атрибутів.

**Висновки.** Розроблено методи пошуку патернів ситуації у вигляді послідовностей подій на трасах логу, а також побудови схеми дій ситуації, що використовує отримані патерни.

Перший метод комбінує використання обмежень по охопленню трасам логу і по частоті появи послідовності подій ситуації у файлі логу. Отримана в результаті застосування методу послідовність подій забезпечує можливість формування графу дій процесу в заданій ситуації.

Другий метод призначений для побудови workflow – схеми ситуації з використанням алгоритму process mining на основі патерну у вигляді послідовності ситуацій, що дає можливість побудувати модель процесу у вигляді послідовності ситуацій із визначенням обмежень для кожної ситуації.

В практичному плані використання запропонованих методів дозволяє адаптувати модель ситуації та процесу в цілому до поточного стану предметної області з використанням відповідних правил вибору дій.

#### Список литературы

1. *Mannila H.* Levelwise search and borders of theories in knowledge discovery / *H. Mannila, H. Toivonen* // *Data Mining Knowledge Discovery*. – 1997. – № 1 (3). – P. 241–258.
2. *Van der Aalst W. M. P.* *Process Mining: Discovery, Conformance and Enhancement of Business Processes* / *W. M. P. Van der Aalst*. – Springer, Berlin Heidelberg, 2011. – 352 p.
3. *Reiter R.* *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems* / *R. Reiter*. – MIT Press, 2001. – 448 p.
4. *Reiter R.* On knowledge-based programming with sensing in the situation calculus / *R. Reiter* // *ACM Transactions on Computational Logic*. – 2001. – № 2 (4). – P. 433–457.
5. *Van Harmelen F.* *Handbook of Knowledge Representation (Foundations of Artificial Intelligence)* / *F. van Harmelen, V. Lifschitz, B. Porter*. – Elsevier Science, 2008. – 1034 p.

#### References (transliterated)

1. Mannila H., Toivonen H. Levelwise search and borders of theories in knowledge discovery. *Data Mining Knowledge Discovery*. 1997. no. 1 (3), pp. 241–258.
2. Van der Aalst W. M. P. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*, Springer, Berlin Heidelberg, 2011. 352 p.
3. Reiter R. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*, MIT Press, 2001. 448 p.
4. Reiter R. On knowledge-based programming with sensing in the situation calculus. *ACM Transactions on Computational Logic*. 2001, no. 2 (4), pp. 433–457.
5. Van Harmelen F., Lifschitz V., Porter B. *Handbook of Knowledge Representation (Foundations of Artificial Intelligence)*, Elsevier Science, 2008. 1034 p.

Надійшло (received) 22.05.2017

#### Бібліографічні описи / Библиографические описания / Bibliographic descriptions

**Побудова ситуаційного представлення знань на основі аналізу логів / С. Ф. Чалий, І. Б. Прибильнова** // Вісник НТУ «ХПІ». Серія: Системний аналіз, управління та інформаційні технології. – Харків : НТУ «ХПІ», 2017. – № 28 (1250). – С. 70–73. – Бібліогр.: 5 назв. – ISSN 2079-0023.

**Построение ситуационного представления знаний на основе анализа логов / С. Ф. Чалий, И. Б. Прибильнова** // Вісник НТУ «ХПІ». Серія: Системний аналіз, управління та інформаційні технології. – Харків : НТУ «ХПІ», 2017. – № 28 (1250). – С. 70–73. – Библиогр.: 5 назв. – ISSN 2079-0023.

**Construction of a situational representation of knowledge based on the analysis of logs / S.F. Chalyi, I.B. Pribylnova** // Bulletin of NTU "KhPI". Series: System analysis, control and information technology. – Kharkov : NTU "KhPI", 2017. – № 28 (1250). – P. 70–73. – Bibliogr.: 5. – ISSN 2079-0023.

#### Відомості про авторів / Сведения об авторах / About the Authors

**Чалий Сергій Федорович** – доктор технічних наук, професор, професор кафедри інформаційних управляючих систем Харківського національного університету радіоелектроніки, м. Харків, тел.: (057) 702-14-51; e-mail: serhii.chalyi@nure.ua.

**Прибильнова Інна Борисівна** – доцент кафедри економічної кібернетики та управління економічною безпекою Харківського національного університету радіоелектроніки, м. Харків, тел.: (057) 702-14-90; e-mail: inna.butukina@nure.ua.

**Чалий Сергей Федорович** – доктор технических наук, профессор, профессор кафедры информационных управляющих систем Харьковского национального университета радиоэлектроники, г. Харьков, тел.: (057) 702-14-51; e-mail: serhii.chalyi@nure.ua.

**Прибильнова Инна Борисовна** – доцент кафедры экономической кибернетики и управления экономической безопасностью Харьковского национального университета радиоэлектроники, г. Харьков, тел.: (057) 702-14-90; e-mail: inna.butukina@nure.ua.

**Chalyi Serhii** – Doctor of Technical Sciences, Full Professor, Professor of the Department of Information Control Systems of the Kharkiv National University of Radioelectronics, c. Kharkiv, (057) 702-14-51; e-mail: serhii.chalyi@nure.ua.

**Pribylnova Inna** – Associate Professor of the Department of Department of Economic Cybernetics and Management of Economic Security of the Kharkiv National University of Radioelectronics, c. Kharkiv, (057) 702-14-90; e-mail: inna.butukina@nure.ua.