S. F. Chalyi, Dc. Sc. (Eng.), Prof.; I. V. Levykin, Cand. Sc. (Eng.), Ass. Prof.

A METHOD FOR BUILDING AN INTERVAL MODEL OF THE CASE-BASED PROBLEM SOLUTION PROCESS BY MEANS OF THE EVENT LOG ANALYSIS

A method is proposed for building an interval model of the case-based problem solution process by analyzing the event log of the process information system. The method includes steps of determining intervals of the log events, which correspond to the process actions, distinguishing the sets of sequential, parallel and independent intervals of the events as well as building the process interval model by joining these sets. Unlike the existing approaches, the signs of sequential or parallel execution are determined not for individual events, but for intervals of events, which makes it possible to represent process actions rather than process states in the model, taking into account duration of performing these actions. Application of the interval model in the framework of the case-based approach enables case selection through the problem solution time estimation.

Key words: *case, business process, process mining, process-based management.*

Introduction

Case-based reasoning (CBR) is directed towards the use of the existing experience for solving new problems [1]. The experience of solving problems is structured in the form of a case [2, 3]. A case comprises characteristic of the problem as well description of its solution process [4]. CBR implementation involves searching and adaptation of the case, its application as well as saving for further use.

The use of case-based reasoning is especially relevant for process-oriented management of an enterprise [1, 3].The cycle of process-oriented management includes the stages of development, improvement and configuring of business process models as well as enterprise management using business processes. Business process comprises algorithm of actions for solving a functional problem, taking into account the available resources. Case-based reasoning implementation in the framework of process-based management creates the possibilities for efficient replication and improvement of business processes.

However, the issues of elaboration of a general approach to case representation in the form of sequences of interrelated actions, taking into account the time aspect, have not been adequately developed, which indicates relevance of the given research.

Analysis of the research and publications

In order to solve process-oriented management problems, process-based information management systems are used. Such systems support solution of functional problems and register execution of the processes in the event log [5].

Modern methods and process mining tools are designed for building models of such processes by finding causal relationships between the events, which are registered in the event log of the information management system [6]. Process analysis methods are intended for building discrete models, which are formalized using mathematical tools of Petri nets, temporal modal logics, process algebra [6, 7]. Models, formed as a result of using such methods, determine the sequence of problem solution steps but do not take into account duration of individual actions, which prevents from solving the problem of searching and selection of an appropriate case for process management problems, taking into account the process duration.

Problem statement

The paper aims at the development of a method for building a case-based model of the problem solution process with interval representation of time. Interval representation of time makes it possible to compare the cases according to the problem solution duration as well as to distinguish intervals of performing actions and expectation of resources.

Practical value of building the model with interval time representation is as follows: in parallel execution of several processes it enables identification of fragments of different processes, which compete for access to the resources, and to organize such access with minimal time delays.

The object of this research are problem solution processes with interval representation of time. Such processes are characterized by the problem solution algorithm as well as by temporal characteristic of the process actions.

To achieve the research aim, the following tasks must be solved:

– to determine the features of sequential, parallel and independent execution of actions in the process event log;

– to develop a method for building the interval model of the problem case-based solution process through analyzing the sequences of events.

A method for interval model construction, based on the event log analysis

Events, which represent execution of the time solution process in the past and are registered in the event log, are used as input data for the method of the interval model construction.

Event log is formed by the information management system for each process of functional problem solution and contains data about the sequence of the process actions. In other words, the log contains information about the problem solution, controlled by the information system.

Each event in the log represents a corresponding process action. Such logs are characterized by registering events for each process sequentially in time as the actions are performed. Sequence of events, which records one process execution from beginning to end is the process trace. Formally, the event log has the following structure:

$$\begin{aligned} &\Pi = \{\pi_k\}, k = \overline{1,K} \\ &\pi_k = \langle E_k, \succ \rangle \\ &E_k = \{e_{k,i}\} \\ &e_{k,i} \succ e_{k,j} \Leftrightarrow e_{k,i} N e_{k,j} \end{aligned}, \quad (1)$$

where Π – event log; π_k – k – process trace; E_k – set of events on the process trace ; $e_{k,i}$ – i – event on the trace π_k ; $\succ$ – transition ratio; N – operator Next of the temporal logic.

The presence of transition ratio $\succ$ between two events $e_{k,i}$ and $e_{k,j}$ means that there are no intermediate events between them, i.e. $e_{k,i} N e_{k,j}$.

In many cases event log realization is performed according to XES standard, which sets the xml-scheme for describing the execution of business processes. The input data structure of this Standard is presented in Fig. 1.

```
<log>
      Definition of variables and attributes of the log
           <trace>
            Set of the trace attributes
            <event>
            Set of the event attributes
            </event>
```

```
                …
            <event>
            List of the event attributes
            </event>
        </trace>
            …
        <trace>
            …
        </trace>
</log>
```

Fig. 1. The method input data structure

While building the model of the problem solution process, it is necessary to compare the same events in different traces of the log. As a rule, however, events in the log do not have an identifier. They are characterized by a set of attributes and their values. As it is evident from Fig. 1, the set of attributes for description of events is defined at the log level. The lists of event attributes are different for different processes. Therefore, assigning unique identifiers to the events should be performed separately for each log, i.e. this is an engineering problem.

Further we will assume that in the input data of the method each unique event has its own identifier. Such identifier makes it possible to determine equivalency of the events, registered on different traces of the log.

The presented formalization of the event log elements makes it possible to determine the interval of performing actions. Such interval must have at least 2 events, which reflect beginning of the action (or completion of the previous operation) and completion of the action. Then the action execution on the process trace we will define as a pair of actions, which are interrelated by the transition ratio

$$\alpha_{k,ij} = [\, e_{k,i}, e_{k,j} \,] \mid e_{k,i} \succ e_{k,j}, \quad e_{k,i}, e_{k,j} \in \pi_k, \tag{2}$$

where $\alpha_{k,ij}$ – interval between boundary events $e_{k,i}$ and $e_{k,j}$ on trace π_k.

If the process action consists of a set of elementary operations, in the log it could be registered as a sequence of several events. Correspondence between the set of events in the log and the process action is determined taking into account the values of the event attributes [8]. These attributes fix the state of action as well as of the objects used in execution of the corresponding action. Commonly, in such processes the log events have the attributes “action name” and “action state”. Said attributes enable distinguishing a subset of events, which corresponds to one action of the process, because the action name will be the same for this subset of events while the state will have different meanings. E. g., the action ‘receiving orders for service” in the service company log could have the following states: waiting, servicing, fulfilled.

If there are several events corresponding to one action, the boundary pair of events $e_{k,i}$ and $e_{k,j}$ is defined through a transitive closure on the transition relation:

$$\alpha_{k,ij} = [e_{k,i}, e_{k,j}] \mid e_{k,i} > e_{k,j}. \tag{3}$$

In this case, between the events $e_{k,i}$ and $e_{k,j}$ there are intermediate events, which correspond to the same action, i.e. $e_{k,i} > e_{k,j} \Leftrightarrow e_{k,i} \succ \ldots \succ e_{k,j}$.

Duration of the interval of events on the trace is determined through the time difference in the occurrence of boundary events of the corresponding action:

$$\Delta\tau_{k,ij} = \tau_{k,j} - \tau_{k,i}, \tag{4}$$

where $\tau_{k,i}$ – the event $e_{k,i}$ occurrence time; $\tau_{k,j}$ – the event $e_{k,j}$ occurrence time.

The main idea of the method for interval model construction consists in determining relations of

sequential, parallel or independent execution between separate actions of the process or the groups of such actions. This approach is the development of $\alpha-$algorithm, where such relations are set for the log events [6, 7].

For the interval model construction relationships between actions are set in the time aspect, which determines relevancy of using temporal modal logic to describe such relationships. It should be noted that logical description of the model provides the possibilities for its further verification in accordance with the ModelChecking paradigm.

For further formalization temporal N operators are used, which determines sequential execution of actions (events) one after another, as well as F operators, which determines sequential execution with intermediate events (actions).

Let us define the time interval, which represents execution of one and the same action on different traces of the process as

$$\alpha_{ij} = [\{ e_{k,i}, e_{k,j} \}] \Leftrightarrow \forall \pi_k (e_{k,i} N e_{k,j} \vee e_{k,i} F e_{k,j}) | \exists (e_{k,i}, e_{k,j} \in E_k), \tag{5}$$

where α_{ij} – interval of events for different traces of the process, E_k – set of events on π_k trace.

From expression (5) it is evident that time interval can be identified by the presence of orderly pairs $e_{k,i} \succ e_{k,j}$ or $e_{k,i} \succ e_{k,j}$ for traces π_k, where events $e_{k,i}$ and $e_{k,j}$ occur.

As a rule, for predicting the time of performing a case-based solution process a maximal and minimal estimates are given. Maximal estimate of the duration of interval α_{ij}, determined on all of the log traces, has the form of

$$\Delta\tau_{ij}^{\max} = \max_k (\Delta\tau_{k,ij}), \tag{6}$$

where $\Delta\tau_{ij}^{\max}$ – maximal estimate of the interval $[e_{k,i}, e_{k,j}]$ duration for all π_k traces.

Minimal estimate is determined in a similar way.

Determination of the event interval (5) makes it possible to formalize the sign of sequential execution of actions as a sequence of event intervals. Conceptually, if actions are performed sequentially, there could be no intermediate events between corresponding intervals. Then, sequential execution is registered on corresponding traces of the process in the form of successive intervals α' and α'' as follows:

$$\alpha' N \alpha'' \Leftrightarrow \alpha' = [e_{k,i}, e_{k,j}] \Rightarrow \alpha'' = [e_{k,j}, e_{k,l}] | \exists (e_{k,i}, e_{k,j}, e_{k,l} \in E_k), \tag{7}$$

where $e_{k,j}$ – common boundary event for both intervals, E_k – set of events of π_k trace.

In accordance with (7), if actions are performed sequentially, the last boundary event of the preceding interval α' on all traces, where these intervals exist, is the first boundary event of the following interval α''.

The process actions could be performed sequentially, but with an interval between them. Such situation often occurs when executors at different levels of organizational hierarchy are involved in handling the process. E. g., after an order for service is accepted, an executor could wait for the chief's consent to purchase components in a chosen company.

Sequential execution of a pair of actions in the process with intermediate actions is determined through the event intervals α' and α'' as follows:

$$\begin{aligned} &\alpha' F \alpha'' \Leftrightarrow \\ &\alpha' = [e_{k,i}, e_{k,j}] \wedge \alpha'' = [e_{k,l}, e_{k,m}] \Rightarrow \alpha''' = [e_{k,j}, e_{k,l}] | \exists (e_{k,i}, e_{k,j}, e_{k,l}, e_{k,m} \in E_k), \end{aligned} \tag{8}$$

where α''' – intermediate interval between intervals α' and α'', $e_{k,i}, e_{k,j}, e_{k,l}, e_{k,m}$ – events, which belong to one trace of the process, E_k – set of events on trace π_k.

Duration of performing the pair of actions α' and α'' is equal to the total duration of performing actions α', α'' and α'''.

Let us consider two typical situations, where parallel or independent processing occurs: splitting or joining the works. In the first case two intervals must have common first boundary event, and in the second case – common last event.

Let us determine *split* ratio between the intervals in the following way:

$$\begin{aligned} &\alpha' \, split \, \alpha'' \Leftrightarrow \\ &\alpha' = [e_i, e_j] \wedge \alpha'' = [e_i, e_m] \mid \exists (e_i, e_j, e_m \in E), E = \bigcup_k E_k , \end{aligned} \tag{9}$$

where α', α'' – intervals that have the same initial and different last boundary events on different traces, E – the set of all events in the log.

Join ratio is determined for intervals having the same last and different initial boundary events on different traces of the process:

$$\begin{aligned} &\alpha' \, join \, \alpha'' \Leftrightarrow \\ &\alpha' = [e_i, e_j] \wedge \alpha'' = [e_l, e_j] \mid \exists (e_i, e_j, e_l \in E), , \end{aligned} \tag{10}$$

where α', α'' – intervals that on different traces have the same last and different initial boundary events, E – the set of all events in the log.

In order to distinguish between parallel and independent execution, it is necessary to formalize the criterion of parallelism. Conceptually, parallelism of the process actions means that there is at least a pair of traces, where given actions are registered in a reverse order:

$$\alpha' \| \alpha'' \Leftrightarrow \alpha' = \exists [e_{k,i}, e_{k,j}] \wedge [e_{s,j}, e_{s,i}] \mid e_{k,i}, e_{k,j} \in E_k, e_{s,j}, e_{s,i} \in E_s, s \neq j) , \tag{11}$$

where α', α'' – intervals on the traces in the log, which correspond to the parallel actions of the process; E_k, E_s – sets of events on different traces of the process.

The method for building the interval model of the case-based problem solution process uses the signs of sequential and parallel execution of actions, presented above.

The method includes the following steps:

Step 1. Building set A of the event intervals α_{ij} for all the process traces in accordance with expression (5).

The necessary condition for this step execution is assigning unique identifiers to each unique event. As it was noted above, each event can be uniquely defined through a set of attributes and their meanings, which are unique for each process.

For convenience we rewrite expression (5) in a more concise form that shows the number of repetitions of each interval on the log traces:

$$\mathrm{A} = \{\alpha_{ij}\}, \alpha_{ij} = [e_i, e_j]^{|\{e_{k,i}, e_{k,j}\}|}, \tag{12}$$

where A – the set of all event intervals in the log; e_i, e_j – boundary events of the intervals without identification of the trace, to which they belong; e_i, e_j – boundary events of the interval on π_k trace; $|\{e_{k,i}, e_{k,j}\}|$ – number of repetitions of the event interval.

At this stage of solving the problem of process duration estimation the set of intervals is complemented with the interval duration values: $\mathrm{A}' = \{\alpha_{ij}, \tau_{ij}\}$.

Step 2. Building the subset of event intervals, which represent pairs of sequential actions of the process in accordance with sign (7).

The set of the pairs of successive event intervals we will also determine, taking into account their number in the log:

$$A^N = \{\alpha' N \alpha''\} = \{[e_i, e_j], [e_j, e_l]^{|\{e_{k,i}, e_{k,j}, e_{k,l}\}|} \Big| |\{e_{k,i}, e_{k,j}, e_{k,l}\}| > 1 \ \}, \quad (13)$$

where A^N – a subset of the pairs of successive intervals; e_i, e_j, e_l – boundary events of successive intervals; $e_{k,i}, e_{k,j}, e_{k,l}$ – boundary events of the successive intervals with trace identification; e_j – boundary event that belongs to both intervals.

Constraint $|\{e_{k,i}, e_{k,j}, e_{k,l}\}| > 1$ shows that sequence of actions should be repeated, i.e. it should be registered at least on two traces of the log.

It is evident that duration of the pair of successive event intervals is equal to the sum of durations of separate intervals.

Step 3. Building a subset of event intervals, which reflect pairs of parallel actions: splitting (14) and joining (15) in accordance with the signs given above:

$$A^{split} = \{\alpha' \underset{split}{\|} \alpha''\} = \{([e_i, e_j], [e_i, e_m])^{|\{e_{k,i}, e_{k,j}, e_{k,m}\}|} \ \}, \quad (14)$$

$$A^{join} = \{\alpha' \underset{join}{\|} \alpha''\} = \{([e_i, e_j], [e_l, e_j])^{|\{e_{k,i}, e_{k,j}, e_{k,l}\}|} \ \}, \quad (15)$$

Step 4. Building a collection of subsets $A^{\#}$ of event intervals, which reflect pairs of independent actions of the process, for which condition (11) is not satisfied. At this stage sets $A^{\#split}$ and $A^{\#join}$ are formed similar to Step 3.

Step 5. Formation of the interval model by establishing relationships between intervals of events from sets A^N, A^{split}, A^{join}, $A^{\#split}$, $A^{\#join}$. Relationships between the intervals are established, if boundary events of both intervals coincide.

Step 6. Complementing the model with transitive successive intervals in accordance with sign (7). At this stage such pairs of sequential actions are determined in the model, between which there are intermediate actions. In further analysis this enables identification of the "bottlenecks" of the process, which lead to execution delays.

Step 7. Complementing the model with time estimations of the event intervals. This makes it possible to estimate the problem solution duration by summing up durations of the intervals for different traces of the model. Said estimates are used for selection of a precedent in case-based reasoning problems.

Let us illustrate realization of the 5 basic steps of the method by the example of the event log, which comprises two traces. As an identifier, we will use the letters of Latin alphabet. A trace will be represented in the form of a tuple of event identifiers.

The input log has the following traces:

{<a,b,c,f,k, g,h,i,j>,

<a,d,e,f,k, g,i, h,j>}.

To illustrate the method steps realization results, both traces were combined and represented in the form of a graph (Fig. 2).

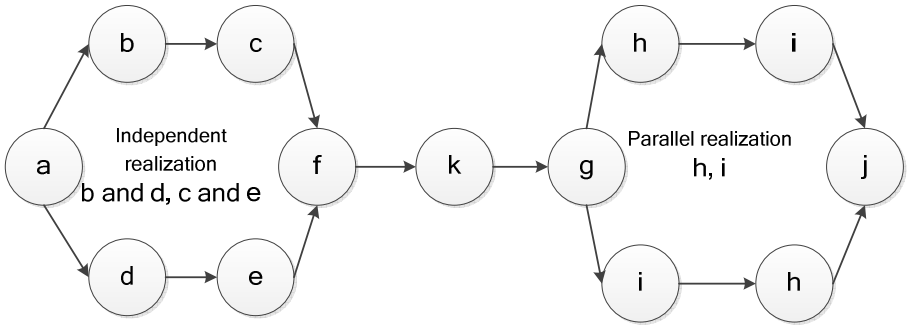

Fig. 2. Joining the events for two traces of the process

The results of Step 1: set of intervals

A = { [a, b], [a, d], [b, c], [d, e], [c, f], [e, f], [f, k], [k, g], [g, h], [g, i], [h, i], [i, h], [i, j], [h, j]}.

The results of Step 2: pairs of sequential intervals $A^N = \{([f, k,][k,g])^2\}$. We have only one pair of sequential intervals, as in this case $|\{ e_{k,i}, e_{k,j}, e_{k,l} \}| > 1$.

The results of Step 3: pairs of parallel intervals A^{split} ={([g, h], [g, i])} and A^{join} = {[i, j], [h, j]}.

It should be noted that for an illustrative example at this stage we did not take into account the number of repetitions, because there are only two traces in the log and each of them registers one variant of parallel execution.

The results of Step 4: pairs of independent intervals $A^{\#split}$ = {[a, b], [a, d]} та $A^{\#join}$ = {[c, f], [e, f]}.

The results of Step 5: sequential joining of the subsets of intervals A^N, A^{split}, A^{join}, $A^{\#split}$, $A^{\#join}$ into a single interval model on the basis of coinciding boundary events:

joining A^N and A^{split}: {([f, k,]–>[k,g]), ([k,g] –> [g, h]), ([k,g] –>[g, i])};

– joining A^N and A^{join}: there are no coinciding boundary events;

– joining A^{split} and A^{join}: {([g, i]–>[i, j]), ([g, h] –> [h, j])}.

Other subsets are joined in a similar way.

Conclusions

The analysis of the event log structure has been performed and the criteria of sequential, parallel and independent execution of the process actions have been determined through the relations between the intervals of log events, which represent execution of the process actions.

A method for building an interval model of the case-based reasoning process of problem solution on the basis of the event log analysis is proposed. The method includes the steps of determining intervals of the log events, which correspond to process actions, distinguishing the sets of sequential, parallel and independent intervals of events as well as building interval model of the process by means of joining these sets.

The method develops the ideas of building a discrete model of the process, which were presented in the process mining alpha-algorithm. Unlike the existing approaches, the criteria of sequential and parallel execution are determined not for separate events, but for intervals of events, which makes it possible to represent the process actions in the model rather than states, taking into account durations of performing these actions.

The interval model application in the framework of the case-based reasoning approach enables case selection on the basis of estimation of the problem solution duration.

REFERENCES

1. Watson I. Case-based reasoning is a methodology not a technology / I. Watson // Knowledge-based systems. – 1999. – № 12. – P. 303 – 308.

2. Николайчук О. А. Применение прецедентного подхода для автоматизированной идентификации технического состояния деталей механических систем / О. А. Николайчук, А. Ю. Юрин // Автоматизация и современные технологии. – 2009. – № 5. – С. 3 – 12.

3. Aamodt A. Case-Based Reasoning: Foundational issues, methodological variations, and system approaches / A. Aamodt, E. Plaza // AI Communications. – 1994. – № 7 (1). – P. 39 – 59.

4. Люгер Д. Ф. Искусственный интеллект: стратегии и методы решения сложных проблем / Д. Ф. Люгер. – М. : Вильямс, 2003. – 864 с.

5. Weske M. Business Process Management: Concepts, Languages, Architectures / M. Weske. [2nd edition]. – Springer-Verlag Berlin Heidelberg, 2012. – 403 p.

6. 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.

7. Van der Aalst W. M. P. Process Mining in the Large: A Tutorial / W. M. P. Van der Aalst // Business Intelligence.

– 2014. – Vol. 172. – P. 33 – 76.

8. Чалый С. Ф. Выявление интервалов ожидания в бизнес-процессах на основе анализа последовательностей событий / С. Ф. Чалый, И. В. Левыкин // Технологический аудит и резервы производства, 2016. – № 5/2 (31). – С. 71 – 76.

Chalyi Serhiy – Dc. Sc. (Eng.), Prof. of the Department of Information Control Systems of Kharkiv National University of Radielectronics, E-mail: serhii.chalyi@national.

Kharkiv National University of Radioelectronics.

Levykin Ihor – Cand. Sc. (Eng.), Ass. Prof. of the Department of Media Systems and Technologies of Kharkiv National University of Radioelectronics, E-mail: ihor.levykin@nure.ua.

Kharkiv National University of Radioelectronics.