Formal Verification of Temporal Constraints for Mobile Service-Based Business Process Models

Deng Zhao, Walid Gaaloul, Wenbo Zhang, Chunsheng Zhu, Zhangbing Zhou

Abstract—Service-based business processes are typically used for achieving business goals through the execution of a set of activities. These activities are usually implemented upon mobile devices in terms of mobile services in a dynamic and pervasive environment. The execution of business processes are recorded in event logs in contemporary enterprise information systems, and are discovered, monitored and improved leveraging process mining techniques. However, process mining from time perspective which has been evolving as an emerging principle for supporting the temporal analysis of these mobile service-based business processes, has not been explored extensively. To address this challenge, we propose to extend the mobile service-based business process model on the basis of the specification and verification of temporal constraints. A timed business process model is proposed where the relative and absolute temporal constraints correspond to each individual activity, sets of activities with control relations, and edges between activities or gateways are estimated. Conformance checking is implemented for the verification and improvement of the proposed model with respect to the accuracy, precision and recall. Extensive evaluations and comparison of our technique with the state of arts are conducted based on publicly-available real-life event logs, and the results demonstrate the effectiveness and verification of the proposed model.

Index Terms—Mobile Service-Based Business Process Model, Temporal Constraint, Conformance Checking.

1 INTRODUCTION

MOBILE service computing has been evolving as a novel paradigm which allows smart things to be represented as mobile services, and to be connected to collaborate, cooperate, and communicate with each other in a dynamic and pervasive environment [1], [2]. Business processes are typically used for satisfying business goals through the execution of a set of activities implemented upon mobile devices and represented in terms of mobile services. For example, a mobile service “replying to customer” which is executed upon a smart phone is involved as an activity in the process as shown in Fig. 2. These processes are usually discovered, monitored and improved, also known as process mining, by extracting knowledge from the event logs commonly available in the information systems [3]. Over the last couple of years many tools and techniques for process mining have been developed [4], [5].

To the best of our knowledge, process mining from time perspective has not been explored extensively. Generally, temporal information is equally important and useful especially for tackling planning and scheduling problems [6], [7]. By using the timestamps in the event log one can extend the model to show bottlenecks, levels of mobile service, throughput times, and frequencies and so on. Specifically, the timestamps in the event log can be derived to analyze the temporal behavior of intra-activity and collaborative activities. Besides, time differences between causally related activities can be extracted from event logs to add possible temporal relations and dependencies between activities or gateways. As a result, temporal analysis and supervision which are usually important for real life systems can be enhanced based on the specification and verification of temporal constraints. Consequently, specifying and verifying the temporal constraints of the mobile service-based business process model is essential and significant for supporting the investigation of the model from time perspective.

Traditional techniques have been proposed for temporal analysis for the business process [8]. In [9] the authors propose a Petri-net based modeling of discrete event systems with temporal constraints where inter-activity temporal constraints are concerned. In [10] the authors describe the main temporal constraints in the management of business process including activity duration, intervals and temporal dependencies between activities and so on. In [11] the authors develop a temporal network representation of an event log for facilitating delay detection. An approach enabling the formal specification and verification of advanced temporal constraints is proposed in [12]. Generally, temporal constraints are considered and formalized for business processes in these works. There are also different works concerning various temporal aspects, such as in [13] the authors give the formal specification of temporal constraints and define the semantics of temporal patterns. A temporal process model based on constraint satisfaction is presented in [14] to check temporal consistency. In [15] the authors provide an automatic mapping of the extended BPMN onto timed automata. In [16] the authors propose to evaluate dynamic temporal constraints based on a temporal distance.

* Deng Zhao and Zhangbing Zhou are with the School of Information Engineering, China University of Geosciences (Beijing), China and the Computer Science Department, TELECOM SudParis, France. Chunsheng Zhu is the corresponding author. E-mail: dengzhao.cugb@gmail.com
* Walid Gaaloul is with the Computer Science Department, TELECOM SudParis, France.
* Wenbo Zhang is with the Technology Center of Software Engineering, Institute of Software, Chinese Academy of Sciences, China.
* Chunsheng Zhu is with the Department of Electrical and Computer Engineering, University of British Columbia, Canada.

*Manuscript received XX XX, 2018; revised XX XX, 201X.*
dependency graph. Generally, the specification and verification of the temporal constraints have hardly been discussed comprehensively for a business process model, and usually are predefined instead of deriving from event logs in real-life applications. Besides, temporal analysis based on the limitedly defined temporal constraints has been researched partially or unilaterally. To remedy this issue, we propose to specify and verify the temporal constraints of intra-activity, inter-activity and collaborative activities, and construct a timed business process model for facilitating temporal behavior analysis. Our contributions are presented as follows:

- Firstly, the temporal constraints including relative and absolute temporal constraints are formally specified. Temporal constraints of individual or collaborative activities are firstly extracted from event logs. Temporal relations and dependencies between activities or gateways are discovered accordingly.
- Then, a timed business process model is constructed which is actually a directed graph where the nodes represent the activities with temporal constraints, the edges indicate the temporal relations and dependencies between the activities, and the gateways refer to the temporal constraints of collaborative activities.
- Finally, conformance checking is applied to verify how compliant the proposed model and a new arrival event log are. Extensive evaluations are conducted based on publicly-available real-life event logs. Experimental performance including the accuracy, precision and recall demonstrates the effectiveness and comprehensiveness of the specification and formalization of the temporal constraints.

The rest of this article is organized as follows. Section 2 acknowledges the temporal constraints. Section 3 introduces the verification of temporal constraints and the discovery of a timed business process model. Then, Section 4 validates the proposed model with conformance checking. The evaluations based on real-life event logs are implemented in Section 5. Section 6 reviews the related works. Finally, Section 7 concludes this work.

## 2 Temporal Constraints

Generally, temporal constraints can be absolute and relative as shown in the following. For more details on the different temporal constraints, we refer the reader to [12].

- **Relative temporal constrains** are used to specify requirements such as activity duration and temporal relation and dependency.

  - **Activity duration**: Given an activity $atv$, it is usually recorded with its start time and finish time (denoted $S(atv)$ and $F(atv)$). Then it is expressed in terms of a time duration $T(atv) = F(atv) - S(atv)$. Considering the minimum and maximum value of the duration, we specify the activity duration in terms of a time interval $[MinT, MaxT]$ with $1 \leq MinT \leq MaxT$.

  - **Temporal relation**: Temporal relation is the constraint crossing the boundaries of activities. Specifically, an activity depends on the start or finish of another activity in order to begin or terminate. Considering temporal relation between two activities $atv_i$ and $atv_j$, $atv_j$ can be executed after, met by, equals, overlapped by, contains, started by, or finished by $atv_i$ [11].

  - **Temporal dependency**: Four kinds of temporal dependencies can be specified according to the start and finish time of activities: Start-to-Start (SS), Start-to-Finish (SF), Finish-to-Start (FS), and Finish-to-Finish (FF). For example, when an activity $atv_a$ is executed before another activity $atv_b$ as shown in Fig. 1, we can calculate the delay between the finish time of $atv_a$ and the start time of $atv_b$, i.e., $FS$ for $atv_a$ and $atv_b$, denoted $FS(atv_a, atv_b)$. On the basis of the temporal relations, the temporal dependencies are analyzed accordingly. Generally, i) if $atv_a$ is executed before, meets or overlaps with $atv_b$, it’s practical to analyze $SF(atv_a, atv_b)$, while ii) $FS(atv_a, atv_b)$ is calculated when $atv_a$ is executed before $atv_b$, and iii) $SS(atv_a, atv_b)$ and $FF(atv_a, atv_b)$ may be concerned when $atv_a$ overlaps $atv_b$, iv) Intuitively, the total execution time for those activities is the maximum activity durations, i.e., $T(atv_b)$, for the rest cases.

- **Absolute temporal constrains** can be specified to define the start and finish times of activities.

  - Must Start On (MSO), Must Finish On (MFO).
  - Start No Earlier Than (SNET) / Finish No Earlier Than (FNET).
  - Start No Later Than (SNLT) / Finish No Later Than (FNLT).

Generally, the absolute temporal constraints are concerned on controlling the start and finish times of activities.
3 Timed Business Process Model

A business process model is usually discovered from event logs where the relationships between a set of activities are defined to achieve a business goal of an organization [17]. Note that an event log is actually a set of events recorded with instance identifier, name, actor, start time and complete time and so on. An example is shown in the Table 1 which corresponds to the example business process model in Fig. 2, the first row means that the activity \( a \) in case 1, i.e., “Get service trouble ticket”, is executed by Marc from 9:05 to 9:06.

<table>
<thead>
<tr>
<th>Case</th>
<th>Activity</th>
<th>Actor</th>
<th>Start</th>
<th>Complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a</td>
<td>Marc</td>
<td>9:05</td>
<td>9:06</td>
</tr>
<tr>
<td>2</td>
<td>b</td>
<td>Marc</td>
<td>9:06</td>
<td>9:20</td>
</tr>
<tr>
<td>3</td>
<td>c</td>
<td>Marc</td>
<td>9:30</td>
<td>12:35</td>
</tr>
<tr>
<td>4</td>
<td>d</td>
<td>Lina</td>
<td>11:46</td>
<td>11:46</td>
</tr>
<tr>
<td>5</td>
<td>e</td>
<td>Lina</td>
<td>12:03</td>
<td>12:15</td>
</tr>
<tr>
<td>6</td>
<td>f</td>
<td>Lina</td>
<td>12:15</td>
<td>12:18</td>
</tr>
<tr>
<td>7</td>
<td>g</td>
<td>Lina</td>
<td>12:16</td>
<td>12:22</td>
</tr>
<tr>
<td>8</td>
<td>h</td>
<td>Marc</td>
<td>12:39</td>
<td>13:10</td>
</tr>
<tr>
<td>9</td>
<td>i</td>
<td>Marc</td>
<td>13:01</td>
<td>13:03</td>
</tr>
<tr>
<td>10</td>
<td>j</td>
<td>Tom</td>
<td>13:02</td>
<td>13:03</td>
</tr>
<tr>
<td>11</td>
<td>k</td>
<td>Tom</td>
<td>13:35</td>
<td>13:36</td>
</tr>
<tr>
<td>12</td>
<td>l</td>
<td>Tom</td>
<td>13:40</td>
<td>13:46</td>
</tr>
</tbody>
</table>

Note that in this setting, a business process model is expressed as a BPMN which is widely adopted for model representation. An example business process model is shown in Fig. 2, where \( d \) and \( e \) are executed exclusively, so does \( f \) and \( g \), \( b \) and \( \{d, e\} \) are executed sequently, as well as \( c \) and \( \{f, g\} \). While \( h \) and \( i \) are executed concurrently. In this paper we propose to extend the business process model with temporal constrains through extracting information from the timestamps in the event logs, and to discover a timed business process model accordingly. A timed business process model is formalized as follows:

**Definition 1 (Timed Business Process Model).** A timed business process model is a tuple \( tbpm = (A, G, E, TC) \) where:

- \( A \) is the set of activities implemented as mobile services.
- \( G \) is the set of gateways between activities.
- \( E \subseteq (A \times A) \cup (A \times G) \cup (G \times G) \) is the set of edges between the activities and gateways.
- \( TC \rightarrow A \cup G \cup E \) specifies temporal constraints to activities, gateways and edges.

Generally, a timed business process model is based on the discovered business process model. Nowadays, there have been various mining algorithms for model discovery [4], and in this setting the inductive miner [18] which is widely adopted for process mining is applied. Based on a discovered business process model, we propose to extract temporal constraints and assign temporal relations and dependencies to activities, edges and gateways:

1) \( TC \rightarrow A \) extracts the possible temporal constraints as introduced in Section 2 correspond to the activities with respect to the start time, finish time and activity duration. Specifically, the start time and finish time correspond to the absolute temporal constraints, where the start time will be constrained by \( SNET, SNLT, \) or \( MSO, \) while the finish time will be constrained by \( FNST, FNLT, \) or \( MO. \) The activity duration as defined in the relative temporal constraints will be formalized with minimum and maximum values, i.e., \( MinT \) and \( MaxT. \) Take the event log shown in Table 1 as an example, the temporal constraints corresponding to the activities are shown in Table 2 and Fig. 2.

2) \( TC \rightarrow G \) specifies the temporal constraints of the gateways. Generally, the gateways usually correspond to exclusive choice (\( \times \)) and concurrency (\( + \)), and come in pairs which specify the structural start point and end point, respectively. As shown in the example business process model, \( d \) and \( e \) are executed exclusively, while \( h \) and \( i \) are executed concurrently. Generally, the set of activities between the pair of gateways (i.e., executed exclusively or concurrently), can be considered as a whole, or a novel activity, where \( SF \) is much valuable to be considered for these collaborative activities in this case. Generally, when a set of activities between a pair of gateways represented as a novel activity, \( TC \rightarrow A \) will extract the corresponding start time, finish time and activity duration as well.

3) \( TC \rightarrow E \) investigates the temporal constraints between the activities and gateways. Firstly, an edge from an activity or a gateway, to another activity or gateway, will be analyzed to which temporal relation it belong by comparing the start time and finish time of the corresponding activities and gateways. There are different kinds of temporal relations as shown in Fig. 1, and the temporal dependencies are analyzed based on the corresponding temporal relations. Note that there is no temporal relation between activities executed exclusively, such as for activities \( d \) and \( e \). An example based on the event log in the Table 1 is explicated in the Table 4. There may be different temporal relations for two activities in different process instances, such as the activity \( a \) meets the gateway \( \{b, d, e\} \) twice and \( a \) before \( \{b, d, e\} \) once in the example event log.

Algorithm 1 elaborates the construction of a timed business process model (denoted \( tbpm \)) based on a given event log (denoted \( elog \)) and a business process model (denoted \( bpm \)) which is discovered by inductive miner in ProM and expressed in BPMN. Firstly, we propose to investigate the temporal constraints of activities, gateways and edges in each process instance (denoted \( t_i \)) in \( elog \) (lines 1-15). Specifically, the total execution time (i.e., \( SF \)) of each gateway (denoted \( gtw \)) are depended on the executions of all activities between \( gtw \) in each trace (lines 2-5). Note that the functions \( min \) and \( max \) return the minimum and maximum values. Note that the set of activities between \( gtw \) is represented as a novel activity and will be put into \( A \) (line 4). Then, the temporal information with respect to each activity (denoted \( atvi \)) is analyzed (lines 6-10). Note that \( atvi \) may be executed for many times in different traces,
TABLE 2
Temporal constraints of activities derived from the example event log in Table 1.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Activity Duration</th>
<th>Finish Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>d</td>
<td>9:30 - 12:25</td>
<td>9:30 - 12:25</td>
</tr>
<tr>
<td>e</td>
<td>12:03 - 12:55</td>
<td>12:03 - 12:55</td>
</tr>
<tr>
<td>h</td>
<td>12:15 - 12:59</td>
<td>12:15 - 12:59</td>
</tr>
<tr>
<td>i</td>
<td>12:06 - 13:01</td>
<td>12:06 - 13:01</td>
</tr>
</tbody>
</table>

TABLE 3
Temporal constraints for the gateways and the corresponding activities.

<table>
<thead>
<tr>
<th>Gateway</th>
<th>Activity</th>
<th>Novel Activity</th>
<th>Temporal Constraints</th>
<th>Start Time</th>
<th>Finish Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>×</td>
<td>d</td>
<td>(d, e)</td>
<td>SF(d, e) = [12h, 3d]</td>
<td>[9:30, 12:03]</td>
<td>[12:15, 12:55]</td>
</tr>
<tr>
<td>+</td>
<td>h</td>
<td>(h, i)</td>
<td>SF(h, i) = [6m, 7m]</td>
<td>[12:15, 12:59]</td>
<td>[12:22, 13:05]</td>
</tr>
</tbody>
</table>

TABLE 4
Temporal relations and dependencies between the activities and the gateways.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Activity</th>
<th>Temporal Relations</th>
<th>Temporal Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>{b, (d, e)}</td>
<td>meets(2), before</td>
<td>FS(a, {b, (d, e)}) = [0, 4m], SF[a, {b, (d, e)}] = [30m, 360m]</td>
</tr>
<tr>
<td>b</td>
<td>{d, e}</td>
<td>before</td>
<td>FS(b, {d, e}) = [1m, 10m], SF[b, {d, e}] = [29m, 349m]</td>
</tr>
<tr>
<td>{b, (d, e)}</td>
<td>(h, i)</td>
<td>meets, before</td>
<td>SF({b, (d, e)}, {h, i}) = [0, 4m], SF(b, (d, e), {h, i}) = [36m, 349m]</td>
</tr>
<tr>
<td>h</td>
<td>i</td>
<td>contains, overlaps</td>
<td>SF(h, i) = [6m, 7m], SS(h, i) = [1m, 2m], FF(h, i) = [3m, 4m]</td>
</tr>
</tbody>
</table>

we propose to analyze the start time (denoted \(at_{v, start}\)), finish time (denoted \(at_{v, end}\)), activity duration (denoted \(at_{v, duration}\)) of each activity. For each edge (denoted \(edge_{i}\)) in \(t_k\) (line 11), the temporal relation is firstly analyzed by \(tmpRep(edge_{i})\) (line 12), based on which the corresponding dependencies are declared as follows (line 13). After traversing all the traces in \(eLog\), the maximum and minimum of the start time, finish time and activity duration are identified accordingly for each individual activity and collaborative activities (lines 16-20). The maximum and minimum value of \(SF\) of each gateway (denoted \(gw_{i, SF}\)) will be found (lines 21-23). So dose those edges where the maximum and minimum values of intervals are calculated accordingly (lines 24-26). Finally, a timed business process model is constructed which is actually a directed graph where the activities, gateways and edges are all attached with temporal constraints.

An example based on the event log is shown in Fig. 2, where each activity is attached with the start time, finish time, activity duration, such as \(b\) is declared to start during [9:06, 13:40], while finish during [9:20, 13:46], with an activity duration between 6 minutes to 14 minutes. The activities with respect to a pair of gateways are represented as a novel activity whose \(SF\) is analyzed accordingly, such as \(d\) and \(e\) are executed exclusively and \(SF[d, e]\) is between 12 minutes to 3 hours and 25 minutes. Note that \(c, f\) and \(g\) are not recorded in the event log, and dismissed in the example which is partial expressed in this setting. The temporal constraints of edges from an activity or a gateway to another activity or gateway are also analyzed. For example, the edge between the activity \(b\) and the gateway \(d, e\) shows that the execution delay is between 1 minute to 10 minutes. Note that the temporal constraints are measured down to the minute in this setting, while it can be measured down to different time slots such as hours or milliseconds according to different applications. Generally, the temporal constraints of the activities shown in Table 2, that of the gateways shown in Table 3 and that of the edges shown in Table 4, are all explicit in the Fig. 2.

4 Conformance Checking

Based on the timed business process model, we are informed of the possible start time, finish time, activity duration of a single or several activities, intervals between activities or gateways. Generally, these temporal information extracted heavily rely on its historical event log. A event log with more data is much beneficial for ensuring its accuracy and effectiveness, regarding to the occasionality of operations or different preferences of actors. Conformance checking is of significance for the verification and improvement of a model’s correctness and reliability from the new arrival event log. In this setting, assuming that the new arrival event log matches the proposed model from control perspective. The differences between the temporal constraints in new arrival event log and that in the model are captured indicating to what extent the model and the new arrival event log match each other.

In addition to considering the lowest and utmost bound of the execution time, such as the earliest and latest start time, i.e., \(SNLT\) and \(SNLT\), the frequencies are also taken into account for the verification. We propose to give a probability analysis instead of a absolute temporal constraints. It’s more practical and appropriate for users to be informed...
Algorithm 1 TimedBPMDiscovey

Require:
- \textit{elog}: an event log
- \textit{bpm}: a business process model discovered by adopting the inductive miner [18].

Ensure:
- \textit{tbpm}: a timed business process model based on \textit{elog} and \textit{bpm}

1: \textbf{for} each trace $t_k \in \textit{elog}$ \textbf{do}
2: \hspace{1em} \textbf{for} each pair of gateway $gt_i \in \textit{tbpm}$ in $t_k$ \textbf{do}
3: \hspace{2em} $gt_i,\textit{SF} \leftarrow \text{add}(\max(gt_i, \text{cpt}) - \min(gt_i, \text{start}))$
4: \hspace{1em} $A$\text{add}novel activity \textit{nativ} $\leftarrow$ activities in $gt_i$
5: \hspace{1em} \textbf{end for}
6: \hspace{1em} \textbf{for} each activity $atv_i \in A \cap \textit{tbpm}$ in $t_k$ \textbf{do}
7: \hspace{2em} $atv_i,\textit{start} \leftarrow \text{add}(atv_i, \text{start})$
8: \hspace{2em} $atv_i,\text{cpt} \leftarrow \text{add}(atv_i, \text{cpt})$
9: \hspace{2em} $atv_i,\textit{duration} \leftarrow \text{add}(atv_i, \text{start})$
10: \hspace{1em} \textbf{end for}
11: \hspace{1em} \textbf{for} each edge $edge_i \in E \cap \textit{tbpm}$ in $t_k$ \textbf{do}
12: \hspace{2em} $edge_i,\text{relation} \leftarrow \text{tmpRep}(edge_i)$
13: \hspace{2em} $edge_i,\text{dependency} \leftarrow \text{add}(SS/\textit{SF}/FS/FF)$
14: \hspace{1em} \textbf{end for}
15: \textbf{end for}
16: \textbf{for} each activity $atv_i \in A \cap \textit{tbpm}$ \textbf{do}
17: \hspace{1em} $atv_i,\textit{SNLT} \leftarrow \min(atv_i, \text{start})$
18: \hspace{2em} $atv_i,\textit{SNLT} \leftarrow \max(atv_i, \text{start})$
19: \hspace{2em} \text{or} $atv_i,\textit{MSO} \leftarrow \text{SNLT}/\text{SNLT} \text{(SNLET==SNLT)}$
20: \hspace{2em} $atv_i,\text{MinT} \leftarrow \min(atv_i, \text{duration})$
21: \hspace{2em} $atv_i,\text{MaxT} \leftarrow \max(atv_i, \text{duration})$
22: \hspace{2em} $atv_i,\textit{FNLT} \leftarrow \min(atv_i, \text{cpt})$
23: \hspace{2em} $atv_i,\textit{FNLT} \leftarrow \max(atv_i, \text{cpt})$
24: \hspace{2em} \text{or} $atv_i,\textit{FMO} \leftarrow \text{FNLT}/\text{FNLT} \text{(FNLET==FNLT)}$
25: \hspace{1em} \textbf{end for}
26: \textbf{end for}

That an activity usually occur at a certain interval with a high possibility (denoted $thrpb$), instead of within the lowest and utmost bound of the start time. Such as instead of knowing $a$ will possibly start during [9:05, 13:35], user may be more convinced of a statement that $a$ is more likely to start during a time interval such as [10:00, 12:00] with a possibility of 80%. Considering the probability distribution of the start time or finish time, and execution time, time is divided with different temporal length with respect to its granularity. In this setting, time is divided to 24 hours for granularity hour, such for the start time and finish time as exampled in Table 5, or divided to 6 ten minutes with granularity minute, while divided to 6 tens seconds with granularity seconds. For example, $b$ is possible to start between [9:06, 13:40], but may usually start during 9:00 to 10:00 AM with a possibility 60%. Note that in this setting $thrpb = 60\%$ and it can be set to other values according to the different requirements.

Generally, the accuracy, precision and recall are calculated for the conformance checking and verification, respectively. Given a new arrival event log $nelog$, the temporal constraints correspond to the activities, gateways and edges are observed firstly. We use the notation $TS_h$ to denote temporal constraints within the expected range with a certain possibility. If the temporal constraints are not within the expected specific range, we use the notation $TS_l$ to denote that within the lowest and utmost bounds. Otherwise, if the observed temporal constraint are not expected, we use the notation $TS_n$ to denote these cases. The accuracy, precision and recall are computed as follows:

\begin{align}
\text{Accuracy} &= \frac{|TS_h + TS_l|}{|TS_h + TS_l + TS_n|} \\
\text{Precision} &= \frac{|TS_h|}{|TS_h + TS_l|} \\
\text{Recall} &= \frac{|TS_l|}{|TS_h + TS_l + TS_n|}
\end{align}
5 IMPLEMENTATION AND EVALUATION

A prototype has been implemented by JAVA, and experiments have been conducted for the evaluation purpose upon a desktop with an Intel(R) Core(TM) i5-2400 CPU at 3.10 GHz, a 4-GB memory and a 64-bit windows system. In the experiments, we firstly give details of the data sets, and then present the performance evaluation and comparison of our technique with other related works.

5.1 Data Sets

The proposed techniques have been evaluated using publicly-available real-life event logs. Instead of private logs which hamper the replicability of the results due to not being accessible, these event logs are access at the 4TU Centre for Research Data [19]. Besides, we consider the activities of daily living of several individuals logs, which includes 8 event logs as shown in Table 5. These logs record the executions of various processes of daily living from different individuals, e.g., read, sleep, outdoors, work, relax and so on which are implemented in terms of mobile services. The log size ranging from 6 (uci_weekends) to 43 traces (104_labor), while the length of a trace vary with traces containing 368 event, to traces containing 4200 events. In this setting, we propose to use 80% data of an event log for the model construction, and the rest 20% data for the conformance checking.

5.2 Performance Evaluation

Considering different event logs in the collection of Activities of daily living of several individuals logs, the average accuracy, precision and recall of the start time, finish time and the duration of activities are shown in Fig. 3 - 5. The average accuracy, precision and recall vary with different event logs. It can be found that the average accuracy of the start times and finish times are high in general, and the precision and recall are around 50% for different event logs, which indicate the effectiveness and applicability of temporal analysis based on the proposed model. While the average precision of the activity duration for different event logs show higher values as shown in Fig. 5, which indicates that the activity duration usually occurred in time intervals with high possibilities as indicated by the history data. On the other hand, the values of average accuracy of the activity duration are lower than that of average precision, which indicates that there are also sometimes that the activity duration may differ from its history data and out of range of the possible minimum and maximum values of the activity duration. From another aspect, we can find that the performance of weekdays is better than that of weekends due to the data sufficiency. Generally, these event logs demonstrate the effectiveness and applicability of our proposed temporal model.

Fig. 6 shows the average accuracy, precision and recall of the execution time of the set of activities between a pair of gateway (i.e, SF) in different event logs, while Fig. 7 shows the average accuracy, precision and recall of the intervals between the activities or gateways (i.e, FS). Generally, the accuracies, precision and recall all are with high values which demonstrate the effectiveness of the temporal analysis based on the proposed model.

Fig. 8 shows the temporal relations of edges in the event logs, where before, meet, overlap, during, start and be finished...
Generally, there are different works which aim to investigate and extend the business process model from the temporal perspective. Such as in [9] the authors propose a Petri Nets based modeling of Discrete Event Systems (PNDES) with temporal constraints where mainly inter-activity temporal constraints are concerned. In [10] the authors describe the main Temporal Constraints in the Management of Business Process (TCMBP) including activity duration, intervals and temporal dependencies between activities. In [11] the authors introduce a Temporal Network Representation (TNR) of event logs. It captures the temporal relations between activities, and the delays between activities. In [15] the authors propose to transform BPMN models to Timed Automata Networks (TAN) where activity durations are formalized using text annotations. In [13] the authors provide precise formal Semantic for Time Patterns (STP) based on different temporal patterns. The work is inspiring and convincing but do to take control relation into account, collaborative activities temporal constraints are not formalized. In [14] the authors introduce the notion of controlled violations of Temporal Process Constraints (TPC) which includes activity duration and inter-activity constraints, i.e., temporal dependency. In [12] the authors propose to enable the formal specification and verification of Temporal Constraints of Business Processes Models (TCBPM) while temporal relations are not included. In [20] the authors propose to extend BPMN with temporal constraint with respect to activity duration, and then map the Extended BPMN (EBPMN) onto timed automata that can be verified by the UPPAAL model checker. In [21] the authors intend to verify BPMN models based on Time Petri Nets (TPN) and formally specify the temporally constrained model where activity duration is mainly concerned. In [22] the authors outline a Time-Interval Genetic Process Mining framework (TIGPM) where time-intervals between events are derived.

The comparative results with our Timed Business Process Model (TBPM) regarding to the temporal constraints (denoted TC) in the related business process models are shown in Table 6. Note that we identified temporal constraints into three categories: i) Intra-activity temporal constraints, ii) Inter-activity temporal constraints and ii) Collaborative activities temporal constraints. From the
comparison work we can find that most of the already studied constraints include the temporal perspective, but they usually consider partial temporal constraints and intend to achieve different temporal analysis goals. However, based on the limited temporal constraints, the temporal analysis such as time conflict analysis is very limited neither. Consequently, our technique in this paper show a more comprehensive specification and verification of temporal constraints in mobile service-based business process model.

6 Related Work

Service-based business process model has been evolving as an emerging principle which sits between service-oriented architecture and process modeling and analysis in a mobile environment [4], [24]. Different techniques have been proposed in recent years, such as in [25], the authors develop a correctness verification approach where various correctness properties for service-oriented business process are identified. In [26] the authors define a formal service-based business process model with the Petri net modelling language, based on which a set of change patterns are identified as primitives for analysing complex changes. In [27] the authors propose an approach to data consistency checking for the dynamic replacement of service process, where service processes are formalized and data consistency problem is discussed. Generally, few mobile service-based business process consider time aspect, in this paper we propose to investigate the temporal constraints based on process mining techniques. Many techniques have been developed to discovery, monitor and enhance business process model by extracting knowledge from event logs readily available in today’s information systems [28]. In [4] the authors provide a systematic review and comparative evaluation of existing automated process discovery methods, using an open-source benchmark and publicly-available event logs. Traditionally, most of these proposals produce BPMN and declarative constraints. However, process mining from time perspective has been relatively unexplored which actually plays an important role with respect to the temporal analysis of the business process model from different temporal aspects.

Generally, specification and verification of the temporal constraints of the mobile service-based business process model have been explored by some researchers [9], [10], [15]. In [8] the authors present a survey on the temporal specification and verification in business processes, where workflows, Web service composition, and inter-organizational domain are considered. Actually, a few research works are related to business process model and as demonstrated in the paper, partial timed properties are considered in different works. Such as in [23] the authors propose to extend BPMN with additional temporal constraints and dependencies between activities. In [15], [20] the authors propose to transform BPMN models to timed automata networks based on the model checking verification techniques. Temporal constraints such as execution time of activities are formalized but predefined using text annotations [15]. In [29] the authors develop a framework of using Petri Net to model timed service business process, where service business process are composed as a service model and extended with time specifications. An automatic timed properties generation in the form of temporal logic formulae is proposed for verification. However, only temporal constraints among business activity interactions are formalized. Besides, the paper focus more on a service process model instead of a business process model. In [30] the authors propose a consistent time-aware cloud resource allocation method where temporal constraints on cloud resources and on process activities are specified in business processes. A temporal process model is proposed in [14] and [31], and different types of temporal constraints are formalized considering the (combined) durations of activities, inter-task interval duration, and overlap among activities. Temporal consistency are checked and an approach is proposed to determine the best schedule to minimize the total penalty from the violations. However, the temporal process model is not discovered based on an event log which is defined and used for temporal consistency checking. In [11] the authors present a temporal network representation (TNR) of an event log for facilitating delay detection and probabilistic variant mining. Interval relations between events are mainly concerned. In [20], [21] the authors concentrate on the activity duration while other temporal constraints have not been discussed when considering the formalization and verification of business processes. While in [22] the authors try to investigate the intervals between events and propose a time-interval genetic process mining framework. Generally, these works are inspiring from time perspective but they

<table>
<thead>
<tr>
<th>Approaches [Ref.]</th>
<th>Intra-activity TC</th>
<th>Inter-activity TC</th>
<th>Collaborative activities TC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Activity duration</td>
<td>Start/Finish TC</td>
<td>Intervals</td>
</tr>
<tr>
<td>FNDES [9]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>TCMBP [10]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>TNR [11]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>TAN [15]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>STP [13]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>TPC [14]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>TCBPM [12]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>EBPMN [20]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>TPN [21]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>TIGPM [22]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Our TBPM</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
do not explore the temporal constraints of business processes comprehensively. On the other hand, these temporal constraints are usually predefined instead of derived from historical event logs. In this setting, the temporal constraints correspond to the mobile service-based business process model regarding to each individual activity, sets of activities with respect to a control relation, and the temporal relations and dependencies between them are comprehensively formalized and analyzed.

7 CONCLUSION
This paper proposes to construct a timed business process model for temporal analysis and supervision, where activities are implemented in terms of mobile services and consist of processes in a mobile environment. Generally, temporal constraints are specified and formalized, and then derived from event logs with respect to the start time, finish time, and execution duration of each individual activity or collaborative activities, and the possible intervals between activities or gateways. A timed business process model is specified according to the specification and verification of temporal constraints. Conformance checking is implemented to verify how compliant the proposed model and a new arrival event log is. Experimental evaluations based on real-life event logs have been conducted, and the results demonstrate that our technique can have a more comprehensive specification and verification of temporal constraints.

ACKNOWLEDGMENTS
This work was supported partially by the National Natural Science Foundation of China (No. 61772479 and 61662021), and by the Fundamental Research Funds for the Central Universities (China University of Geosciences (Beijing), China).

REFERENCES