International Journal of Multimedia and Ubiquitous Engineering
Vol.9, No.5 (2014), pp.273-280
http://dx.doi.org/10.14257/ijmue.2014.9.5.27

A Temporal Reasoning based Social Calendar Framework

Yuechang Liu[1] and Yong Tang[2]

[1]Jiaying University, South China Normal University, [2]South China Normal University
ychangliu@gmail.com, ytang@m.scnu.edu.cn

Abstract

Electronic calendar has been intensively used in people's life and work. In recent years it is popularized in the context of social networks, which forced the emergence of social calendar. For social calendar people is not just satisfied with creating, storing and exchanging their schedule data, they wants to use the calendar as a social network app, send invitations on some scheduled events, flexibly define the access rules on their scheduling, and even create event agenda in team manner. This paper describes a framework to answer those requirements. Using the standard interoperaterable calendar data format - iCalendar as calendar representation, the framework compiles the event information into an internal network structure, and incorporates a MapReduce consistency algorithm - that makes the framework suitable for large scale data processing in social network context.

Keywords: *Social Calendar, Temporal Reasoning, MapReduce*

1. Introduction

In modern era electronic calendar (including events and activities) data is everywhere. Take a postgraduate student for example, he take courses that is scheduled in the college teaching affairs system and can be obtained from ***MyFriday***[1](with Chinese name 超级课程表); he shares his personal schedule on ***Doodle***[2] where his friends can find the appropriate time to meet him; he can browses and discovers some interesting events from Internet and decides to attend; when in the lab he discusses about the project he participates in with other team members; before he decides to go to a scheduled academic conference, he adds that conference event to his to-do list and makes a whole plan to write a paper for that conference, etc. Indeed, electronic calendar system has been a key means of personal time management and team collaboration. It is intensively used in areas like personal scheduling, course timetable planning, conference/meeting arrangement, and project management.

Once upon a time, people only use the electronic calendar tool to edit, show, share, interoperate and exchange calendar data. When used in social network context (like ***MyFriday*** and ***Doodle*** mentioned above), the situation is somewhat different. Like other social network applications, user experience is a key feature. Ideally, the system is expected to tell people whether his newly added event is consistent with his future plans with his friends; the system recommends some events to people that interests him and he happens to have time for it; the system computes feasible free time intervals for users' scheduling decision; even the system can compute possible solutions for team events' scheduling in the context of privacy protection (*e.g.* The system automatically computes a best time for a team meeting in according to every member's personal time table). All of the mentioned features can be implemented using different temporal reasoning algorithms.

ISSN: 1975-0080 IJMUE

To investigate the problems, let's consider following running example. Suppose Ann and her friend Bill have registered on a same SNS calendar site, and they have each created a personal schedule on the same day. According to the schedule Ann will have a 60 minute excise before spending 90 to 120 minutes performing a physical therapy to her injured foot (the therapy is prescribed by Bill); Bill will spend 90-120 minutes prescribing a physical therapy for Ann and drop it off, and then he is about to giving a lecture in the university from 10:00 to 12:00. In this scenario, following problems may be concerned by users: Can I export my calendar data and import it to my company MS Outlook server? Can Ann attend Bill's lecture in time? When users accept the collaborative events from his friend, can the system find and remind him whether it is conflict with his personal agenda? What time can be scheduled in this week for me to join the part? Can anyone derive any private schedule information which is not made to be public?

To develop successful social calendar, there are some key factor: the representation of calendar data that supports interoperation, appropriate efficient reasoning algorithm and data access security. Calendar data representation has been relatively well studied, which produces the iCalendar RDF specification [1], which is regarded as de facto standard for web calendar. The iCalendar is also adopted for the calendar representation in this paper. This paper will be mainly dedicated to the first and second factor, i.e. the incorporation of the appropriate reasoning mechanism into iCalendar standard. The remaining of this paper is organized as follows: the framework is first described in the next section; followed by the detailed description of reasoning mechanism in section 3. The last section is dedicated to related and future work.

2. Framework

The basic framework provides two main features: First, it uses iCalendar as data interoperational protocol; Second, it provides large-scale automated reasoning algorithms for inconsistency detection and message reminding. It is illustrated as Figure 1.

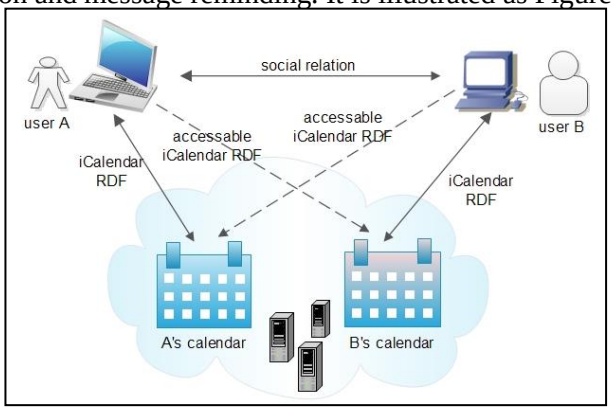

Figure 1. Framework Illustration

Suppose both A and B are users of a social calendar and, they are identified as friend of each other (which means that their calendar data are visible to each other). Both A and B can create new events of his own or related to another one on their schedules. When they create a new event related to another one, e.g. A invites B to take part in a conference and B accepts it, the event will be added to their schedules. Before B accepted the invitation, the system is responsible for giving reminding message about wether the addition of the events will cause

any conflicts on their schedule or not. Or the system can even give advice or recommendation about to whom and when users send invitations.

2.1. Calendar Representation

Consider again the example mentioned above. First, Ann and Bill fill their agenda in a SNS calendar website. The system stores the data in RDF format which is partly depicted in Fig. 2. When they successfully create a piece of event, the server of the framework will run algorithms to update an internal temporal data structure and return according notice to the user end.

Ann's calendar data:

```
<vevent>
  <Vevent>
    <description>Exercing</description>
     <dstart>
      <Date rdfs:label="2013-12-20">
       <hour>8</hour>
       <minute>0</minute>
      </Date>
     </dstart>
    <dend>
      <Date rdfs:label="2013-12-20">
       <hour>9</hour>
       <minute>0</minute>
      </Date>
     </dend>
     </Vevent>
<Vevent>
    <uid rdf:resource="palm:Appointments/77" />
     <description>Physical Therapy</description>
      <dstart>
       <Date rdfs:label="2013-12-20">
        <hour>9</hour>
        <minute>0</minute>
       </Date>
      </dstart>
     <dend>
       <Date rdfs:label="2013-12-20">
        <hour>10</hour>
        <minute>30</minute>
       </Date>
      </dend>
  <X-duration-min>90</X-duration-min>
  <X-duration-max>120</X-duration-max>
</Vevent>
 </vevent>
...
```

Bill's calendar data:

```
<vevent>
  <Vevent>
    <description>Therapy Planning</description>
     <dstart>
      <Date rdfs:label="2013-12-20">
       <hour>8</hour>
       <minute>0</minute>
      </Date>
     </dstart>
    <dend>
      <Date rdfs:label="2013-12-20">
       <hour>9</hour>
       <minute>0</minute>
      </Date>
     </dend>
  <X-duration-min>90</X-duration-min>
  <X-duration-max>120</X-duration-max>
     </Vevent>
<Vevent>
    <uid rdf:resource="palm:Appointments/787" />
     <description>Lecture</description>
      <dstart>
       <Date rdfs:label="2013-12-20">
        <hour>9</hour>
        <minute>0</minute>
       </Date>
      </dstart>
     <dend>
       <Date rdfs:label="2013-12-20">
        <hour>10</hour>
        <minute>30</minute>
       </Date>
      </dend>
</Vevent>
 </vevent>
...
```

Figure 2. Ann and Bill's Calendar RDF

On Figure 2, the uncertainty of events' duration is recorded by extended properties X-duration-min for the minimum value of duration and X-duration-max for the maximum value. Being parsed by iCalendar parser, the temporal information of the data is passed to the central data structure for temporal information management unit - Temporal Information Processor (TIP), which is described as follows.

2.2. Temporal Information Processor (TIP)

TIP is central unit for temporal information management. A large-scale matrix (called Temporal Matrix, TM for short) is utilized to maintain the information, each entry of which records the ***temporal distance*** between two respective ***time points***. When people creates a new event, two time points respect to the events are added into TM, one of which represents the start time of the event and the other for the end. And then, TIP will run temporal consistency algorithm to update TM.

Take the running example for consideration, after Ann and Bill created their schedule, TIP adds respective time points into TM, the local TM is depicted in Figure 4.

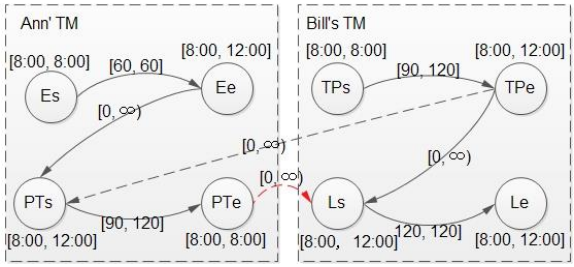

Figure 3. Illustration of TMs for Ann and Bill

In Figure 3 each time point is represented by a vertex, and the edges between time points are the precedence relationship between them, with possible gap (quantity in minutes) in the attached intervals (for the edge a->b tagged with [i, j), its corresponds to the inequation $i \leqslant b-a<j$). In the picture, Es and Ee respectively stand for start and end time of excising, PTs and PTe for those of physical therapy, TPs and TPe therapy planning, and Ls and Le for lecture, respectively. The relation between PTe and Ls tagged with [0, ∞) (the dashed arrow in red) reflects Ann's intention of going to Bill's lecture timely.

The duration constraint on time points essentially constitute **Simple Temporal Problem (STP)** defined in [2] and the respective TM graph is usually called **Simple Temporal Network (STN)** [2].

With this TM in memory, there is much useful temporal information can be derived. For example, when Ann wants to attend Bill's lecture (establish an link between PTe and Ls with gap [0, ∞)) the system will compute and return with an alerting message like "it is not possible for Ann to take the lecture in time."

3. Algorithms

As mentioned above, the framework relies heavily on powerful and efficient reasoning capacity. The TP compiles each schedule event into time points and duration constraints, and maintains the information with the TM structure. Though there have been much research result on STP, they cannot be applied straightforward to the context of this paper. The main reason for this is the application background of this research: central data structure and large amount of data for social calendar, which means quite large scale of the problem. For the MapReduce [3] design pattern has been the factual standard for big data processing. This paper also seeks to design the reasoning algorithm in MapReduce manner. The consistency decision is basic requirement for our application context, here we mainly discuss about it. Before introducing the MapReduce-mannered algorithm, basic Floyd-Warshall (FW) algorithm and Directive Path Consistency (DPC) algorithm are first introduced in following text.

3.1. Directed Path Consistency Algorithm

DPC algorithm is based on usual Floyd-Warshall (FW) algorithm. Compared to based FW algorithm, DPC saves more iteration (that permits better efficiency), while loses the feature of decomposibility [2]. When only consistency decision required, DPC is more desirable. Both FW and DPC are described as Figure 5 and Figure 6 respectively. From the description, DPC relies on a given order that regulates the update order of temporal distance between time points. From the observation that triangle structure on STNs is quite useful for deriving more efficient algorithm, the order in DPC is usually select the triangulation order of STNs (more information on STN triangulation please see [4]).

Algorithm Floyd-Warshall:

Input: A fully-connected distance graph $\mathcal{G} = \langle V, E \rangle$
Output: A FPC distance graph $\mathcal{G}$ or INCONSISTENT

```
1: for k = 1...n do
2:   for i = 1...n do
3:     for j = 1...n do
4:       w_ij ← min(w_ij, w_ik + w_kj)
5:       if w_ij + w_ji < 0 then
6:         return INCONSISTENT
7:       end if
8:     end for
9:   end for
10: end for
11: return G
```

Figure 4. Floy-Warshall Algorithm

Algorithm Directed Path Consistency:

Input: A triangulated temporal network $\mathcal{G} = \langle V, E \rangle$ and corresponding elimination order $o = (v_1, v_2, \ldots, v_{n-1}, v_n)$
Output: A DPC distance graph $\mathcal{G}$ or INCONSISTENT

```
1: for k = 1...n do
2:   for all i < k s.t. e_ik ∈ E do
3:     for all j < i s.t. e_jk ∈ E do
4:       w_ij ← min(w_ij, w_ik + w_kj)
5:       w_ji ← min(w_ji, w_jk + w_ki)
6:       if w_ij + w_ji < 0 then
7:         return INCONSISTENT
8:       end if
9:     end for
10:  end for
11: end for
12: return G
```

Figure 5. Directed Path Consistency Algorithm

3.2. Directed Path Consistency Algorithm in MapReduce (MRDPC)

MapReduce was first proposed by Google [3] and became more and more popular for its simplicity in parallel computing manner. Though not all consistency algorithm can be designed in MapReduce manner, DPC is suitable for MapReduce-style implementation. Just split the range of k into two subranges: k1∈ 1..n/2 and k2∈ n/2+1 .. n. By observation it is found that each w_{ij}(or w_{ji}) is only possibly updated according to the comparison of itself and w_{ik}+w_{kj}, (or w_{jk}+w_{ki}), this expression is completely different for k1 and k2. The fact implies that the value of w_{ij}/w_{ji} can be independently computed for the two subranges. Then, the mappers and reducers are designed as Figure 6.

Mappers:
Input: <key = network name, value = TD>
Output: <key = row index i, value = d_{i*}>
Reducers:
Input: <key = row index i , value = d_{i*}>
Output: <key = row index i, value = updated d_{i*}>

Figure 6. Mappers and Reducers for MRDPC

Considering the running example, the initial local TM is a 9*9 matrix which is illustrated as Figure 7. In Figure 7, a special temporal variable z is used as reference time point for other time points (here we suppose the reference point is 8:00 am on 20th Dec. 2013).

Now suppose that Ann records in her calendar that she wants to take Bill's lecture in time. That makes the system intend to insert the entries PTe-Ls and Ls-PTe with 0 and -∞. Then, by running MRDPC algorithm a negative cycle will be found (TPe->PTs->PTe->Ls->TPe), which the inconsistency is detected. Ann would receive a prompt message that "You are not supposed to finish the event in time".

$$
\begin{array}{c}
\begin{bmatrix} z & Es & Ee & PTs & PTe & TPs & TPe & Ls & Le \end{bmatrix} \\
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 60 & \infty & \infty & \infty & \infty & \infty & \infty \\
-240 & -60 & 0 & 0 & \infty & \infty & \infty & \infty & \infty \\
-240 & \infty & -\infty & 0 & 90 & -\infty & \infty & \infty & \infty \\
0 & \infty & \infty & -120 & 0 & \infty & \infty & \infty & \infty \\
0 & \infty & \infty & \infty & \infty & 0 & 90 & \infty & \infty \\
-240 & \infty & \infty & 0 & \infty & -120 & 0 & 0 & \infty \\
-240 & \infty & \infty & \infty & \infty & \infty & -\infty & 0 & 120 \\
-240 & \infty & \infty & \infty & \infty & \infty & \infty & -120 & 0
\end{bmatrix}
\end{array}
$$

Figure 7. Local TM for running example

3. Conclusions, Related and Future Work

In this paper a temporal reasoning based social calendar framework is proposed. The framework can be applied to social calendar to provide suitable prompt information in context, which helps to improve the user experience for that kind of tools. A running example is used for illustration of the using of the framework. Moreover, the authors give their observation that traditional directed path consistency for STP consistency decision is quite suitable to MapReduce solving phylosophy, and propose a MapReduce based Directed Path Consistency algorithm (MRDPC). MRDPC makes it possible to solve large scale social calendar data.

iCalendar is a factual standard for interoperable electronic calendar data[1] that is adopted by Apple and Google. Based on iCalendar, Lisa Dusseault and E. James Whitehead developed an open calendar sharing and scheduling framework – CalDAV [5]. It is well-recognized that social calendar is trend of electronic calendar application [6]. To obtain better user experience, it is better introduce temporal reasoning capacity to social network, which is the underlying requirement for the proposed framework of this paper. Simple Temporal Problem (STP) is a powerful and suitable temporal model for this paper. Once STP was proposed by Rina Dechter [2], it has been continuously studied for past two decade. Efficient algorithms for STP has widespread application in temporal reasoning, and it is usually adopted as core component in planning and scheduling. Besides Dechter's DPC, Lin Xu proposed an efficient partial path consistency algorithm for STP - Δ STP in 2003 [7], which is further improved by Planken's P^3C algorithm in 2008[8]. Based on STP model, Luke Hunsberger proposed a new temporal model: Temporal Decoupling Problem (TDP) and relevant algorithm [9]. To utilize TDP in robots collaboration, Boerkoel James further formalized the TDP with shared and private STP and proposed a series of centralized and

distributed TDP algorithms [10]. TDP is appropriate model for group time management, particularly in social network context where information is usually tagged as private or shared and accessed in controlled, such that it is a feasible well-defined model for social calendar application.

In the near future, other STP algorithms in MapReduce manner will be explored, including the algorithm design and detailed experiment evaluation. The incorporation of TDP into the framework will also be investigated. Social calendar for team collaboration is another interesting and meaningful research topic for the next step.

Acknowledgement

The research is supported by National Science Foundation Project (No. 60970044 and 61272067) and Laboratory Management Foundation Project of Guangdong High Education Society (No. GDJ2012044).

References

[1] iCalendar (Internet Calendaring and Scheduling Core Object Specification) Standards, official web site http://www.ietf.org/rfc/rfc2445.txt
[2] Dechter, R., Meiri, I. & Pearl, J. Temporal Constraint Networks, Artificial Intelligence 49, pp. 61-95, (1991)
[3] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," in SDI'04: Sixth Symposium on Operating System Design and Implementation, San Francisco, CA, (2004).
[4] Christian Bliek, and Djamila Sam-Haroud, Path Consistency on Triangulated Constraint Graphs. IJCAI, page 456-461. Morgan Kaufmann, (1999).
[5] Chris Lord, Evolution of the electronic calendar: introducing social calendaring, (2008)
[6] Lisa Dusseault, and E. James Whitehead Jr.. Open Calendar Sharing and Scheduling with CalDAV. IEEE Internet Computing 9(2):81-89 (2005)
[7] Lin Xu, Berthe Y. Choueiry, A New Efficient Algorithm for Solving the Simple Temporal Problem. 01/2003; In proceeding of: 10th International Symposium on Temporal Representation and Reasoning / 4th International Conference on Temporal Logic (TIME-ICTL 2003), (2003)
[8] P3C: A New Algorithm for the Simple Temporal Problem. Léon Planken, Mathijs de Weerdt, and Roman van der Krogt. ICAPS, page 256-263. AAAI, (2008)
[9] Luke Hunsberger: Algorithms for a Temporal Decoupling Problem in Multi-Agent Planning. AAAI/IAAI 2002:468-475, (2002)
[10] Distributed Approaches for Solving Constraint-based Multiagent Scheduling Problems. Boerkoel Jr, James C. (2012)

Authors

Yuechang Liu, he received his Ph. D. in computer software and theory, postdoctoral researcher in School of Computer Science, South China Normal University. His research interests include knowledge engineering, social computing and temporal reasoning.

Yong Tang, he is a Professor and Ph.D. supervisor in computer science in School of Computer Science, South China Normal University. His research interests include temporal database, social computing and computer supported collaborative work.

