On Applying Point-Interval Logic to Criminal Forensics

(Student Paper)

Mashhood Ishaque
Abbas K. Zaidi
Alexander H. Levis

Command and Control Research and Technology Symposium
**Report Documentation Page**

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204. Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

<table>
<thead>
<tr>
<th>1. REPORT DATE</th>
<th>2. REPORT TYPE</th>
<th>3. DATES COVERED</th>
</tr>
</thead>
<tbody>
<tr>
<td>JUN 2006</td>
<td>00-00-2006 to 00-00-2006</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4. TITLE AND SUBTITLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>On Applying Point-Interval Logic to Criminal Forensics (Briefing Charts)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5a. CONTRACT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>5b. GRANT NUMBER</td>
</tr>
<tr>
<td>5c. PROGRAM ELEMENT NUMBER</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5d. PROJECT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>5e. TASK NUMBER</td>
</tr>
<tr>
<td>5f. WORK UNIT NUMBER</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6. AUTHOR(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>George Mason University, System Architecture Laboratory, 4400 University Drive, Fairfax, VA, 22030</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</th>
</tr>
</thead>
<tbody>
<tr>
<td>George Mason University, System Architecture Laboratory, 4400 University Drive, Fairfax, VA, 22030</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>8. PERFORMING ORGANIZATION REPORT NUMBER</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>10. SPONSOR/MONITOR’S ACRONYM(S)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>11. SPONSOR/MONITOR’S REPORT NUMBER(S)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>12. DISTRIBUTION/AVAILABILITY STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approved for public release; distribution unlimited</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>13. SUPPLEMENTARY NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>The original document contains color images.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>14. ABSTRACT</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>15. SUBJECT TERMS</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>16. SECURITY CLASSIFICATION OF:</th>
<th>17. LIMITATION OF ABSTRACT</th>
<th>18. NUMBER OF PAGES</th>
<th>19a. NAME OF RESPONSIBLE PERSON</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. REPORT unclassified</td>
<td>b. ABSTRACT unclassified</td>
<td>c. THIS PAGE unclassified</td>
<td>32</td>
</tr>
</tbody>
</table>

*Standard Form 298 (Rev. 8-98)*
Prescribed by ANSI Std Z39-18
Outline

- Formal Logic
- A Logic for Time
- Point-Interval Logic
- Point Graphs
- Temper – Software Implementation of Point-Interval Formalism
- Applying Point-Interval Logic to London Bombing Data
Logic is the “Art of Reasoning”.
Logic is used to make inferences based on the available information.
Formal logic makes inferences based purely on the form of the content, without any understanding of the meaning of the content.
Reasoning based just on the form is important because this means computers can do it.

All humans are mortal.
Socrates is a human.
Therefore Socrates is mortal.

∀x \ (x \in h \rightarrow x \in m)
\ s \in h
\ ?
A logic for time will enable us to:

- Characterize time-sensitive attributes of a domain to be modeled
- Do temporal analysis of a domain, which will help us in developing a better understanding of the relationship between domain entities
- To identify inconsistencies and anomalies

There are very few approaches that allow explicit representation of time and reasoning about it.
Allen’s Interval Logic

- Allen introduced Interval Algebra as a framework for temporal reasoning. The algebra takes time intervals to be primitives. There are 13 possible relationships between a pair of intervals:

\[ R = \{ \text{before, meets, overlaps, starts, during, finishes, equals, after, met-by, started by, contains, finished by} \} \]

- A temporal relation is represented as ‘Xi Cij Xj’, where Cij \( \subseteq R \), and Xi and Xj are intervals. The elements in the set Cij form a disjunctive temporal constraint on relationships between the two intervals.

- The problem of determining consistency in Allen’s Interval Algebra is NP Complete.
Point-Interval Logic (PIL) is a Pointisable logic. It is a tractable subclass of Allen’s interval logic.

Case I: X and Y both intervals with non-zero lengths

\( X = [sx, ex], Y = [sy, ey] \) with \( sx < ex \) and \( sy < ey \)

<table>
<thead>
<tr>
<th>Relation</th>
<th>X ( \leq ) Y</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before</td>
<td>( X &lt; Y )</td>
<td>( ex &lt; sy )</td>
</tr>
<tr>
<td>Meets</td>
<td>( X \simeq Y )</td>
<td>( ex = sy )</td>
</tr>
<tr>
<td>Overlaps</td>
<td>( X \cap Y )</td>
<td>( sx &lt; sy; sy &lt; ex; ex &lt; ey )</td>
</tr>
<tr>
<td>Starts</td>
<td>( X \preceq Y )</td>
<td>( sx = sy; ex &lt; ey )</td>
</tr>
<tr>
<td>During</td>
<td>( X \preceq Y )</td>
<td>( sx &gt; sy; ex &lt; ey )</td>
</tr>
<tr>
<td>Finishes</td>
<td>( X \succeq Y )</td>
<td>( sy &lt; sx; ey = ex )</td>
</tr>
<tr>
<td>Equals</td>
<td>( X = Y )</td>
<td>( sx = sy; ex = ey )</td>
</tr>
</tbody>
</table>
### Point-Interval Logic

**Case II:** X and Y both points

X = [px] and Y = [py] with sx = ex = px and sy = ey = py

<table>
<thead>
<tr>
<th>Relation</th>
<th>Condition</th>
<th>Illustration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before &lt; Y</td>
<td>px &lt; py</td>
<td><img src="image" alt="Before X &lt; Y" /></td>
</tr>
<tr>
<td>Equals X = Y</td>
<td>px = py</td>
<td><img src="image" alt="Equals X = Y" /></td>
</tr>
</tbody>
</table>

A point-point relation “less than or equal to” (≤) can be added to PIL without losing tractability.

**Case III**—X is a point and Y is an interval: X = [px] and Y = [sy, ey] with px = sx = ex and sy < ey

<table>
<thead>
<tr>
<th>Relation</th>
<th>Condition</th>
<th>Illustration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before X &lt; Y</td>
<td>px &lt; sy</td>
<td><img src="image" alt="Before X &lt; Y" /></td>
</tr>
<tr>
<td>Starts X s Y</td>
<td>px = sy</td>
<td><img src="image" alt="Starts X s Y" /></td>
</tr>
<tr>
<td>During X d Y</td>
<td>sy &lt; px &lt; ey</td>
<td><img src="image" alt="During X d Y" /></td>
</tr>
<tr>
<td>Finishes X f Y</td>
<td>px = ey</td>
<td><img src="image" alt="Finishes X f Y" /></td>
</tr>
<tr>
<td>Before Y &lt; X</td>
<td>ey &lt; px</td>
<td><img src="image" alt="Before Y &lt; X" /></td>
</tr>
</tbody>
</table>
Point-Interval Logic

- Quantitative Temporal Information
  - \(d_1 \leq \text{Length} [X, Y] \leq d_2\)
  - \(t_1 \leq \text{Stamp} [X] \leq t_2\)

where \(d_1, d_2, t_1, \) and \(t_2\) are rational numbers, and \(X, Y\) are points.
Point Graphs

PIL Statements and the corresponding Point Graph

p1 < p2
p2 ≤ p3
Stamp p1 = 4500
Length[p1, p2] = 100
Temper

System Architectures Laboratory

viking.gmu.edu
Modeling with Temper

- Convert the available temporal information into statements in Point-Interval Logic.
- Input these statements to Temper using the language editor of Temper.
- Construct a Point Graph representation of the set of PIL statements.
- If the set of PIL statements is inconsistent then Temper will not be able to construct the Point Graph representation.
- Temper will identify the subset of PIL statements causing the inconsistency.
- Remove the inconsistent statements.
- Once a consistent Point Graph has been constructed, it can be used to draw inferences.
Information regarding events unfolds in no specific order
Temporal information may be both qualitative and quantitative
Information may be inconsistent/incorrect
Information may contain hidden patterns or temporal relations that can help identify missing links
An automated tool for temporal knowledge representation, verification and reasoning is required
Example: London Bombing

- There were four explosions in London.
- The sites of these explosions were: Travistock Square, Edgware Road, Aldgate and Russell Square.
- Three of these explosions (Edgware, Aldgate and Russell Square) were in trains.
- These trains left from King's Cross station. The journey of these trains ended in explosions.
- The time it takes a train from King's Cross to reach Edgware is at least 5 time units.
- The time it takes a train from King's Cross to reach Aldgate is at least 4 time units.
- The time it takes a train from King's Cross to reach Russell Square is at least 5 time units.

\[
\begin{align*}
Interval & \text{ Train\_King\_Cross\_to\_Edgware, } \\
& \text{ Train\_King\_Cross\_to\_Aldgate, } \\
& \text{ Train\_King\_Cross\_to\_Russell\_Sq } \\
Point & \text{ Explosion\_at\_Travistock\_Square, } \\
& \text{ Explosion\_near\_Edgware, } \\
& \text{ Explosion\_near\_Aldgate, } \\
& \text{ Explosion\_near\_Russell\_Sq } \\
Explosion\_near\_Edgware & \text{ finish } \\
& \text{ Train\_King\_Cross\_to\_Edgware } \\
Explosion\_near\_Aldgate & \text{ finish } \\
& \text{ Train\_King\_Cross\_to\_Aldgate } \\
Explosion\_near\_Russell\_Sq & \text{ finish } \\
& \text{ Train\_King\_Cross\_to\_Russell\_Sq } \\
Length & [\text{Train\_King\_Cross\_to\_Edgware}] \geq 5 \\
& [\text{Train\_King\_Cross\_to\_Aldgate}] \geq 4 \\
& [\text{Train\_King\_Cross\_to\_Russell\_Sq}] \geq 5
\end{align*}
\]
query Stamp (when did the train to Edgware leave from King’s Cross?)
query Stamp (when did the train to Edgware leave from King’s Cross?)
Example: London Bombing (cont’d)

- The explosion near Edgware Road took place between time units 840 and 852.
- The explosion near Aldgate took place between time units 845 and 850.
- The explosion near Russell Square took place between time units 840 and 850.
- The explosion at Travistock Square took place between time units 945 and 955.

840 \leq \text{Stamp [Explosion near Edgware]} \leq 852
845 \leq \text{Stamp [Explosion near Aldgate]} \leq 850
840 \leq \text{Stamp [Explosion near Russell Sq]} \leq 850
945 \leq \text{Stamp [Explosion at Travistock Square]} \leq 955
revised Point Graph
query Stamp (when did the train to Edgware leave from King’s Cross?)
query Stamp (when did the train to Edgware leave from King’s Cross?)

upper bound = 847
adding a new PIL statement
inconsistency detected
inconsistency identified
deleting a PIL statement
revised Point Graph (consistent)
Example:
London Bombing (cont’d)

- The alleged four bombers spotted entering the Luton station at time unit 720.
- The next train from Luton to King's Cross left at 748 reaching King's Cross at 842.
- The three trains from King's Cross station in which the explosions took place, must have left King's Cross after the train from Luton reached King's Cross.

Interval Train_Luton_to_King_Cross
Point Bombers_spotted_at_Luton
Stamp [Bombers_spotted_at_Luton] = 720
Stamp [sTrain_Luton_to_King_Cross] = 748
Stamp [eTrain_Luton_to_King_Cross] = 842
eTrain_Luton_to_King_Cross < Train_King_Cross_to_Edgware
eTrain_Luton_to_King_Cross < Train_King_Cross_to_Aldgate
eTrain_Luton_to_King_Cross < Train_King_Cross_to_Russell_Sq
the Point Graph
query Stamp (when did the train to Edgware leave from King’s Cross?)
query Stamp (when did the train to Edgware leave from King’s Cross?)

upper bound = 847
lower bound (strict)= 842
Summary of Approach

- Convert the available temporal information into statements in Point-Interval Logic.
- Input these statements to Temper using the language editor of Temper.
- Construct a Point Graph representation of the set of PIL statements.
- If the set of PIL statements is inconsistent then Temper will not able to construct the Point Graph representation.
- Temper will identify the subset of PIL statements causing the inconsistency.
- Remove the inconsistent statements.
- Once a consistent Point Graph has been constructed, it can be used to draw inferences.
Questions?