



jTLEX (TimeLine EXtraction)

A Method For Extracting Timelines
From TimeML Temporal Graphs

Capstone 1: Luis Robaina, Emmanuel Garcia, Pedro Diaz

Capstone 2: Adrian Silva, Christopher Eberhard

Product Owners: Mark Finlayson, Mustafa Ocal

Mentor: Mustafa Ocal

Instructor: Masoud Sadjadi

Meet Our Team!

Christopher Eberhard



Adrian Silva



Meet Our Product Owner/Coaches

Prof.
Masoud
Sadjadi
(Instructor)



Prof. Mark
Finlayson



Mustafa
Ocal





The Problem

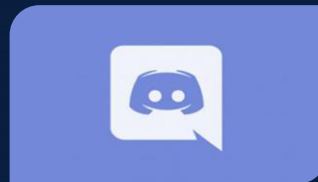
- The current implementation of TimeML outputs annotations on text. Generated temporal graphs partially reveal only as much information such as ordering of events and times. For many applications, this process is inadequate due to its abstraction of data and lack of full ordering approach

The Solution

- jTLEX (TimeLine EXtraction) is a new approach that uses generated temporal graphs to create exact ordering of events and times. This approach is based on prior work that derived a few methods but constructs a new data structure that users can use to understand and visualize ordering events in a better and more intuitive way.

Technologies We Used

- List of technologies, programming languages, testing frameworks, external packages/libraries, collaboration and communication tools

The JUnit logo, featuring the word "JUnit" in a stylized font. The "J" is green and the "Unit" is red.

jTLEX – Constructing and Partitioning Graphs

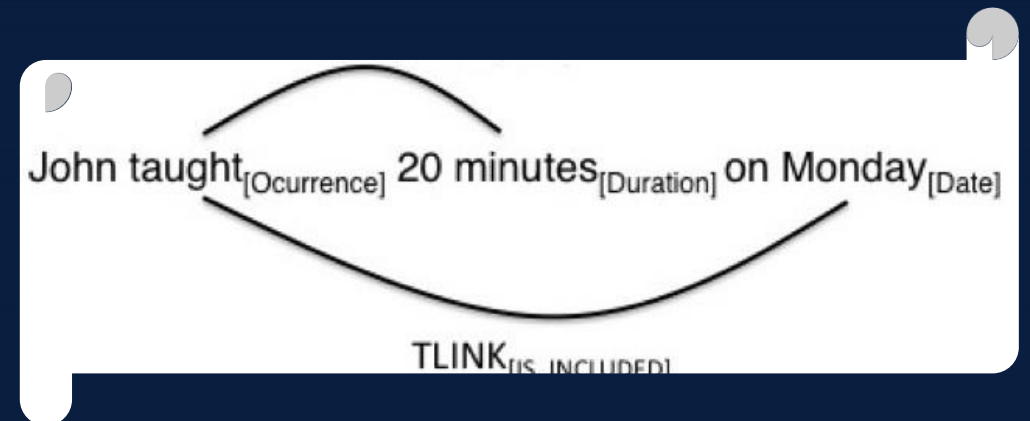
- Contributors:
 - **Adrian Silva**
 - **Emmanuel Garcia**
 - **Luis Robaina**
- Sprint 2:
 - Creation of the base graph data structure
 - Creation of the partition class
 - Creation of the subclasses for subgraphs

Context On Graph And Subgraph Structure

There are 3 Tags we used to identify three different temporal relations:

- TLINKS (Temporal Links) 14 Types
 - Represent time ordering relations between events and times
- SLINKS (Subordination Links) 6 Types
 - This link is to preserve the accuracy of the timeline of events
 - Identifies events that did not occur in our timeline
- ALINKS (Aspectual Links) 5 Types
 - This links marks the initions, continuation and ending of events

Image shows the annotations of events and times using these links

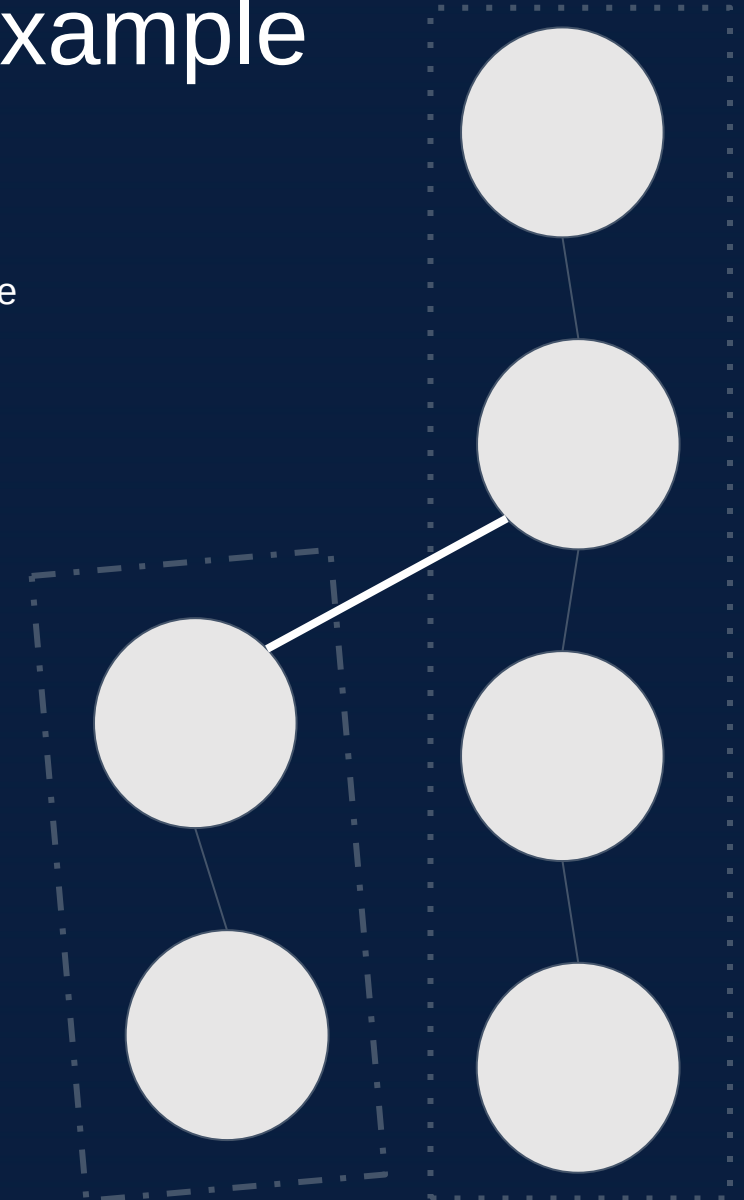


MainGraph And SubGraphs Example

A graph is composed of nodes which represent the events taken place and the edges of a graph are the temporal links

A main graph is composed all of all temporal links including SLINKS

A subgraph is composed exclusively of SLINKS



jTLEX – Constructing and Partitioning Graphs

Part 2

- Contributors:
 - **Adrian Silva**
 - **Emmanuel Garcia**
 - **Luis Robaina**
- Sprint 3:
 - Refactor function for depth first search
 - Write test cases and compare output

jTLEX – Writing Converter For TimeML to TCSP

- Contributors:
 - **Luis Robaina**
 - **Christopher Eberhard**
 - **Pedro Diaz**
- Sprint 3:
 - Transform graph nodes to jacop elements
 - Transform each graph ILink to jacop constraints
 - Assign integer values to the jacop problems
 - Create junit tests for the methods

jTLEX – Creating The Timeline Data Structure

- Contributors:
 - **Adrian Silva**
 - **Emmanuel Garcia**
 - **Luis Robaina**
 - **Pedro Diaz**
- Sprint 4:
 - Adding and searching for breaking points
 - Adding iterator interface
 - Implement timeline class
 - Write test cases

jTLEX – Creating The Inconsistency Detection Class

- Contributors:
 - Emmanuel Garcia
 - Luis Robaina
- Sprint 5:
 - Consistency detector interface
 - Return inconsistencies subgraph
 - Implement detection method
 - Get inconsistency types
 - Write test cases

jTLEX – Creating The Timeline Data Structure

- Contributors:
 - Emmanuel Garcia
 - Luis Robaina
- Sprint 5:
 - Add gettimeline method
 - Add getsubordinatedtimeline method
 - Write test cases
 - Fix timeline length method

jTLEX – Implement Indeterminacy Calculator

- Contributors:
 - **Luis Robaina**
 - **Emmanuel Garcia**
 - **Christopher Eberhard**
- Sprint 6:
 - Create interface for indeterminacy calculator
 - Create function for returning indeterminate time points
 - Create junit tests for the methods