



**Engineering
& Computing**
School of Computing
& Information Sciences



**Knight Foundation
School of Computing
and Information Sciences**

*Florida International University
Knight Foundation School of Computing and Information Sciences*

ANAnSI (Advanced Narrative Analytics System Infrastructure)

Team Members:

Anthony Nguyen - Capstone 2
Cristian Caro - Senior Project
Daniel Chang - Senior Project
Joshua Jimenez - Senior Project
Joshua Posada - Senior Project
Pablo Maldonado - Capstone 2

Product Owners: Dr. Mark Finlayson, Dr. Karine Megerdooomian, Mustafa Ocal.

Project Manager: Antonela Radas

Instructor: Dr. Masoud Sadjadi

Table of Contents

Introduction.....	5
Current System.....	5
Purpose of New System	5
User Stories	6
Implemented User Stories	6
Pending User Stories	6
Project Plan	7
Hardware and Software Resources	7
Sprints Plan	8
Sprint 1	8
Sprint 2.....	8
Sprint 3.....	8
Sprint 4.....	8
Sprint 5.....	8
Sprint 6.....	8
System Design	9
Architectural Patterns.....	9
Deployment Diagram.....	9
Installation Guide.....	9
Short Comings	11
Glossary	12
References	12

Abstract

This document presents the information necessary to gain a good understanding of how the jTLEX java library and ANAnSI conversion is intended to work and what the goal of our work on this project was.

The jTLEX java library was intended to provide a means of taking in a TimeML file which normally contains tags which represent times, events, instances, and links within a document and to use the library to be able to take the TimeML documents and parse it so that the tags can be read to create a timeline from the TimeML which can be easier to read and understand for the end user. It is intended that there would be a java library with the methods necessary in order to allow for the users to make use of the libraries for themselves as a means of inputting any TimeML documents and get an easy to read and understand timeline out of the document. Part of this conversion involves parsing ANAnSI graph output and storing it to then be converted into TimeML standards.

The MIT License (MIT)

Copyright (c) 2016 Florida International University

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

INTRODUCTION

The ANAnSI prototype or Advanced Narrative Analytics System Infrastructure was developed by MITRE. It can generate several levels of analysis that are built on top of each other, these levels include language, event, and temporal analysis. The jTLEX java library is a library created with the intention of allowing for people to download the jTLEX library and make use of the library in conjunction with TimeML files to provide the user with a simpler and easier to read timeline of events that were listed in the TimeML file. The intention of extracting the timeline from the TimeML file was to create methods that would make objects out of the tags and information within said tags and take that information and print out the events provided in an order in which they occurred as to allow insight as to the timeline of how the event went down. In collaboration with MITRE, to achieve conversions to TimeML standards, ANAnSI graph output must be processed and then converted to satisfy this requirement.

Current System

The purpose of the ANAnSI is to map the ANAnSI data structure to TimeML structure for Timeline extraction. The current system is the backend code which parses through a json file and extracts the information needed to create the ANAnSI data structure, once the data structure is created with the information extracted, the system maps the ANAnSI data structure to TimeML data structure. Once the mapping is done, the system will do the timeline extraction and print some output.

Purpose of New System

The purpose of the new system is to be able to take in ANAnSI graph output, via json files, to then be parsed, stored, and mapped to TimeML output.

USER STORIES

The following section provides the detailed user stories that were implemented in this iteration of the ANAnSI(jTLEX) project. These user stories served as the basis for the implementation of the project's features. This section also shows the user stories that are to be considered for future development.

Implemented User Stories

- Data Structure for capturing/holding ANAnSI information
 - Part of language classes
 - Part of language class testers
- ANAnSI annotation parser
- ANAnSI system adaptor
- Implementing TLink section of the adaptor

Pending User Stories

- Neo4j graph database parser

PROJECT PLAN

This section describes the planning that went into the realization of this project. This project incorporated the agile development techniques and as such required the sprints to be planned. These sprint planning are detailed in the section. This section also describes the components, both software and hardware, chosen for this project.

Hardware and Software Resources

The following is a list of all hardware and software resources that were used in this project:

- Desktops/Laptops
- Discord
- Eclipse
- Java JDK
- Java 14
- Trello
- Subclipse
- SVN

Sprints Plan

Sprint 1

- Reviewed Timeline Extraction and required reading
- Download and Install Eclipse and Subclipse
- Download Java JDK 14

Sprint 2

- Write java documentation for each class within data structure
- Write JUnit5 test class to ensure class is fully prepared for conversion
 - Getter methods
 - Proper Input
 - Constructor Initialization

Sprint 3

- Implement Json parser for ANAnSI annotations
- Implement JUnit5 test class to ensure information is processed correctly

Sprint 4

- Write the adaptor to ANAnSI system

Sprint 5

- Do Neo4j (native graph database) Course

Sprint 6

- Write TLink for the adaptor
- Finish Parser
- Write code which will transform Neo4j into ANAnSI data structure.

SYSTEM DESIGN

We started by separating the individual node types into its respective Class and Interface. The JsonParser then parse the document into a set of methods which store Linguistic Events, Clauses, Relations, Temporal, Compound Term and Terms into a SET of Interface Nodes Array Objects with its properties. The Adaptor then stores that information to their TimeML counterpart.

Architectural Patterns

jTLEX:ANAnSI is a java library and as such does not have any software architecture of its own, it is composed entirely of code written in java and is intended to be run through java code editors.

Deployment Diagram

As the jTLEX:ANAnSI java library was incomplete by the time the final sprint was reached we had not discussed any deployment plans in depth, as we were more focused on discussing possible user stories which needed to be implemented prior to us considering how we intended for the project to be deployed.

INSTALLATION GUIDE

Download and install:

1. Java JDK 14

Java SE Development Kit 14.0.2

This software is licensed under the [Oracle Technology Network License Agreement for Oracle Java SE](#)

Product / File Description	File Size	Download
Linux Debian Package	157.93 MB	 jdk-14.0.2_linux-x64_bin.deb
Linux RPM Package	165.06 MB	 jdk-14.0.2_linux-x64_bin.rpm
Linux Compressed Archive	182.06 MB	 jdk-14.0.2_linux-x64_bin.tar.gz
macOS Installer	176.37 MB	 jdk-14.0.2_osx-x64_bin.dmg
macOS Compressed Archive	176.79 MB	 jdk-14.0.2_osx-x64_bin.tar.gz
Windows x64 Installer	162.11 MB	 jdk-14.0.2_windows-x64_bin.exe
Windows x64 Compressed Archive	181.56 MB	 jdk-14.0.2_windows-x64_bin.zip

2. Download Eclipse 2020 - 06/09

Eclipse IDE 2020-06 R Packages



Eclipse IDE for Java Developers

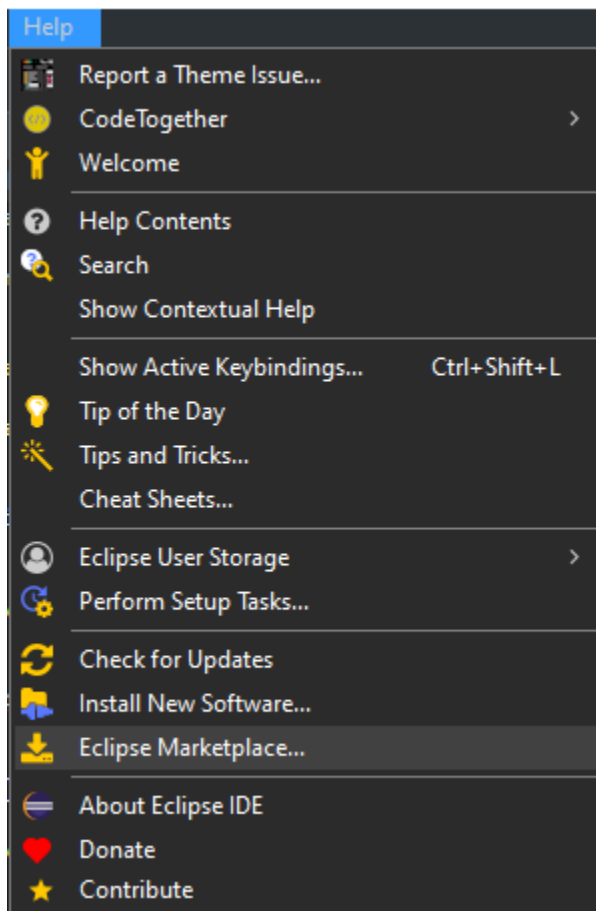
208 MB
1,416,364 DOWNLOADS

The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Maven and Gradle integration



Windows x86_64
macOS x86_64
Linux x86_64

3. Download Subclipse 4.3.0 through Eclipse Marketplace



Subclipse 4.3.3

 An Eclipse Team Provider plug-in providing support for Subversion within the Eclipse IDE. Developed and maintained by Subversion core committers, Subclipse is... [more info](#)

by [Subclipse Project](#), EPL
[svn](#) [subversion](#) [team](#) [provider](#) [mylyn](#) [alm](#)

★ 2205  Installs: 2.73M (14,117 last month) Installed

SHORT COMINGS

The team was no able to finish the Neo4j due to confidentiality issues and permissions. Since we could not create the database in Neo4J, the team was not able to work on the parser.

Also, there were issues with the output as we wanted to change it to see the individual sentences to see their individual links, and instances for easier checking, but we had issues with changing the classes. Therefore, MITRE and another team is going to continue working on this project in the future.

GLOSSARY

Neo4j: Native graph database, built from the ground up to leverage not only data but also data relationships. Neo4j connects data as it is stored, enabling queries never imagined, at speeds never thought possible.

Timex: Time expression.

TimeML: Time Markup Language.

TLEX: Timeline extraction.

SVN: SubVersion tool for Version control. It is open source written in Java. Software developers use Subversion to maintain current and historical versions of files such as source code, web pages, and documentation.

REFERENCES

J. F. Allen (1983) Maintaining Knowledge about Temporal Intervals

Kenneth S. Rubin. Essential Scrum: A Practical Guide to the Most Popular Agile Process

Ocal, M., & Finlayson, M. (2020). TLEX: A Formally Correct Method for Extracting Exact Timelines from TimeML Temporal Graphs

Pustejovsky et. al. (2003) The TimeBank Corpus

Saurí et. al. (2006) TimeML Annotation Guidelines