Project Title: Generalized Language Abstraction and Specification System (GLASS)

Team Members:
Tommy Galletta (tgalletta2022@my.fit.edu)
Alexander Lockard (alockard2022@my.fit.edu)

Faculty Advisor/Client:
Dr. Stansifer (ryan@fit.edu)
Florida Institute of Technology, Department of Computer Science

---

Current Milestone Progress Matrix:

Task	Completion %	Tommy	Xander	Todo
Investigate tools	85%	40%	45%	Investigate other parser generators
Hello world demo	85%	45%	40%	Parser generator demos
Resolve technical challenges	60%	30%	30%	Continued research
Requirements Document	100%	80%	20%	
Design Document	100%	20%	80%	
Test Plan	100%	80%	20%	

Task Discussion:

Investigate tools:

- Choosing a programming language: Java, Rust, Python, and C++ were all investigated and evaluated based on a variety of factors in order to decide which programming language would be best for writing the project in. Ultimately, Java was chosen.
- Comparing existing parser generator tools: This task we decided to hold off on a little bit, as we felt it was more important to get the core functionality of the parser generator working first, before investigating tools and determining the best structure for our syntax specification format.
- Finding XML tools to use: Several XML tools were investigated, and we have settled on a set of viable tools that we will pick from to use. Technical issues may prevent us from using certain tools, which is why we choose to keep our options open for the moment.
- Finding GUI tools to use: Similar to above, we have decided on a set of tools that would be good options for creating our GUI, but we have decided to push back the creation of the GUI in our original plan as making the core functionality is more important to handle first. We will be choosing which GUI tool to use at a later date, but have noted the viable options.

Hello world demos:

- Programming language demos: Demo programs were created in Java, Rust, Python, and C++ for testing speed and memory usage for each programming language in several different use cases.
    - The results of these demo tests can be found here
    - The source code for all demos made can be found here
- Parser generator demos: Similar to above, this investigation will simply occur at a later date.
- XML demo: Small demos were created using several different XML tools to test how effectively these tools could be used to manipulate XML in a variety of ways.
- GUI demo: As a test of a few GUI tools, some basic demos were made containing, a button, a label, and some basic drawings on a canvas

Resolve technical challenges:

- Defining syntax specification format: A decent amount of discussions and research occurred around this topic during Milestone One, and while we have a rough idea of what we want the syntax specification files to look like, we feel that until we fully cement the format of these files, this task should remain unresolved.
- Syntax specification to parse tree: This basically boils down to making a parser for the syntax specification files. This is currently in the works, with great progress already made, and should be done soon.
- Designing the macro-based refactoring tool: This task was left mostly untouched, besides texting XML manipulation. The set of functions we will allow within the macro systems is still yet to be determined. Though will be determined by the end of the semester.

Requirements document:

- Created requirements document, which outlines functional, interface, and performance requirements, among others.

Design document:

- Created design document, which includes a UML diagram for the entire GLASS pipeline, as well as a mock-up sketch of the GUI we plan to make.

Test plan:

- Created test plan document, which includes a list of tests that we plan to perform to help identify bugs, and tests we will perform to evaluate our tool in categories such as intuitiveness and memory efficiency.

Team Member Contributions:

Tommy Galletta:

- Researched and compared programming languages (Java, Rust, Python, C++).
    - This included writing several test programs in each of the four programming languages to test things such as memory and time efficiency in a variety of scenarios.
- Researched parsing algorithms including LL, LR, and LALR algorithms to decide on which parsing algorithm we want to use.
- After LR(1) was decided on, began work on making the LR(1) parser generator Has so far implemented calculations for first sets, closures, and transition states.
- Worked on requirements, design, and test plan documents

Alexander Lockard:

- Investigated several GUI tools to find the best options for creating the GUI tool in the future.
    - This included making a small demo using each tool of a GUI containing basic things such as a button, label, and canvas with lines drawn on it.
- Researched parsing algorithms including LL, LR, and LALR algorithms to decide on which parsing algorithm we want to use.
- Created a small demo of potential XML manipulations that may be implemented as functions for the macro system.
- Worked on requirements, design, and test plan documents

Milestone Two Plan:

Task	Tommy	Xander
Parser generator intermediary checkpoint	Implement core parser generator features	Test and demo parser generator
Investigate other parser generators	Investigate 2-3 parser generator tools	Investigate 2-3 parser generator tools
Solidify syntax specification format “version one”	Groups members will work together in order to ensure all necessary features are implemented	
Implement, test, and demo XML output	Test and demo XML output	Implement core XML manipulations for macros

---

Discussion of Planned Tasks:

Parser generator intermediary checkpoint:

- By the end of Milestone Two, our parser generator should be mostly working. The syntax specification file format may be very clunky by this point, but that will be resolved during future tasks. Also, care will be taken to ensure that all components within the code are written in a modular fashion.

Investigate other parser generators:

- We want to take some time to investigate other parser tools, see what they get "right" and "wrong", what features in other tools feel intuitive and what feels clunky or overcomplicated. We will investigate 4-6 tools, and will document the pros and cons of their different features.

Solidify syntax specification format "version one":

- Now that the strengths and weaknesses of other parser generator tools have been investigated, we will move into designing the first "official" version of the syntax specification format we wish to use in our tool. We will use the research done on the syntax specification formats in other tools as inspiration for the structure of our syntax specification files.

Implement, test, and demo XML output:

- Being able to parse a user-defined programming language is only half the battle. By the end of Milestone Two we hope to also output an XML file containing the original source code inputted by the user, marked up in a tree structure based on the structure of the user-specified grammar for the language.

Client Feedback on Current Milestone:

- See Faculty Advisor Feedback below

Milestone One Faculty Advisor/Client Meeting Dates:

- January 31st
- February 7th
- February 14th

Faculty Advisor Feedback:

Investigate tools:

- Advisor agreed that Java was a good choice of programming language. Not only are we already very familiar with the language, but the advisor also feels that time efficiency is probably the most important thing to focus on with this tool.

- Advisor emphasized the importance of investigating other parser generator tools to find what “works” and what “doesn’t work” within each tool before we solidify the design for the components users will have to write themselves, such as the syntax specification format.
- Advisor was not incredibly familiar with the tools we had mentioned for XML handling, but recommended that we look into tools such as XSLT either for inspiration for our own implementation or to utilize in the creation of our project.

Hello world demos:

- Some of the demos created during this milestone were simply for research purposes (such as the programming language demos) and were not shown directly to the advisor. However, two demos were shown to the advisor.
- A demo of the first, closure, and transition functions for an LR(1) parser was shown to the advisor, who was quite pleased with the results.
- A demo of basic XML manipulation was shown to the advisor, which led into the discussion of looking into tools such as XSLT

Resolve technical challenges:

- Advisor agreed that the details of how the syntax specification files and macros are formatted aren’t incredibly important to solidify this early in the project.
- As for researching the process of going from a syntax specification and source code file to a XML parse tree, the advisor is still a little confused on our motivation, but is nonetheless pleased with the research and progress made thus far.

Faculty Advisor Signature: _____ Date: _____

Evaluation by Faculty Advisor

Please detach and return this page to Dr. Chan (HC 209) or email the scores to pkc@cs.fit.edu

TG = Tommy Galletta
AL = Alexander Lockard

TG	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
AL	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10

Faculty Advisor Signature: _____ Date: _____