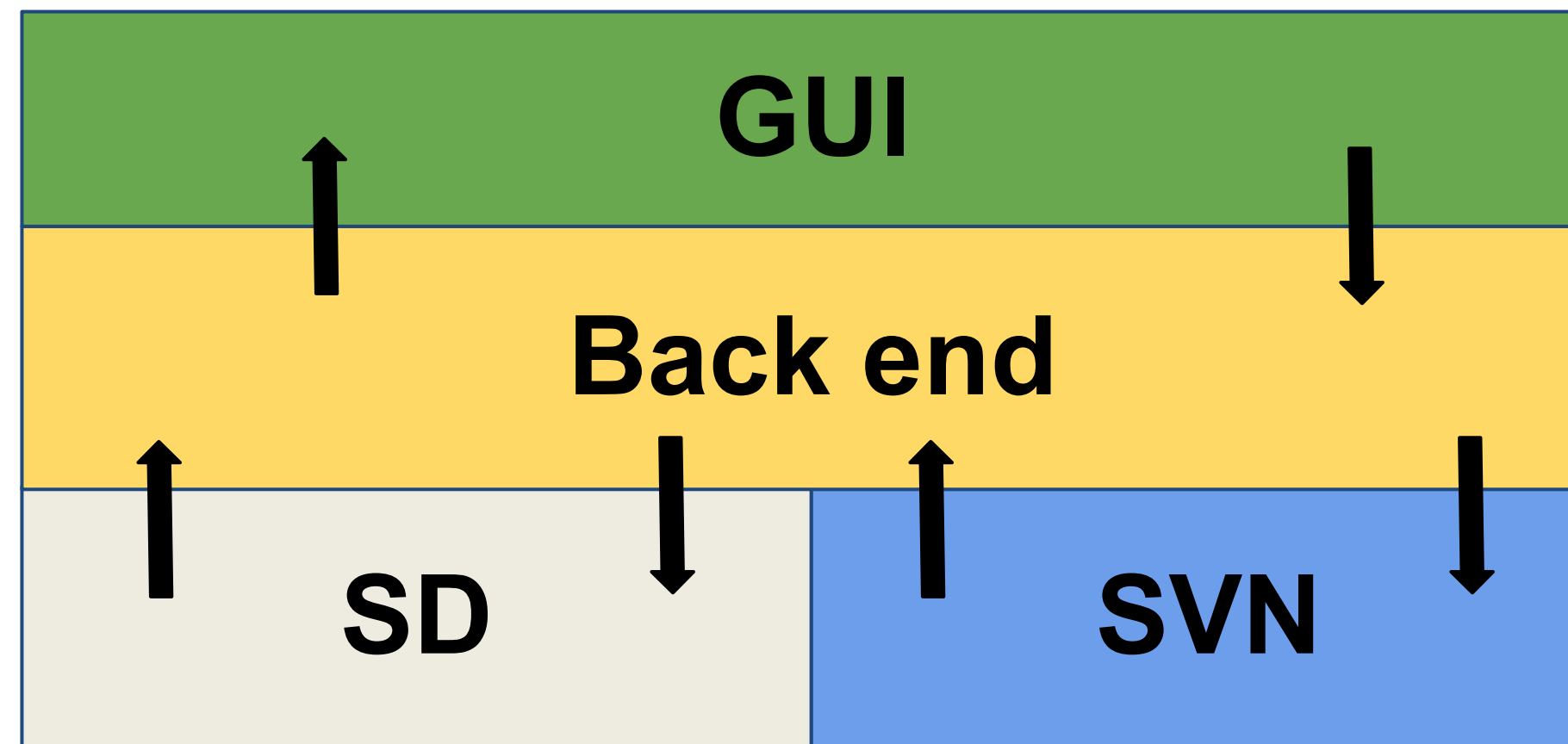


## INTRODUCTION

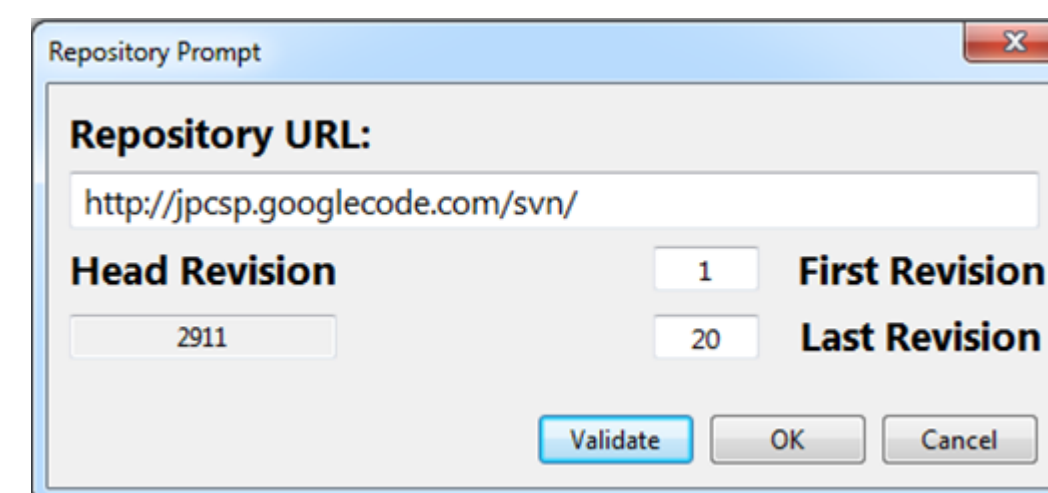
Lexical Search is a textual search that is aware of the syntax of a language. This enables more intelligent queries. For example: one can search for "aim" in a program and specify that only the results which are function names should be returned. Alternatively: in a document one could specify that only nouns are wanted.

A Version Control System is a management tool which tracks changes in computer programs, documents, and other sets of information which can be related to specific versions and other sets of metadata.

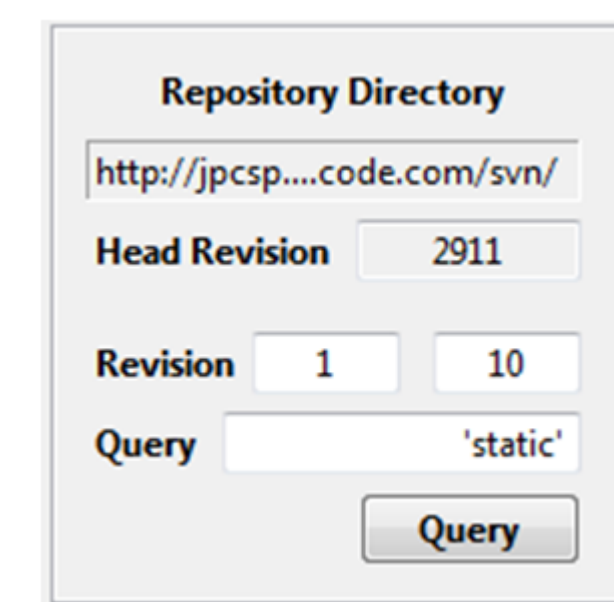
Combining the above two ideas together into a proof of concept was the focus of this project. The synthesis of a Version Control System with Lexical Search yields the ability to not only obtain a meaningful result, but to also see the evolution of that result over time and its relationship to other components of the larger system.



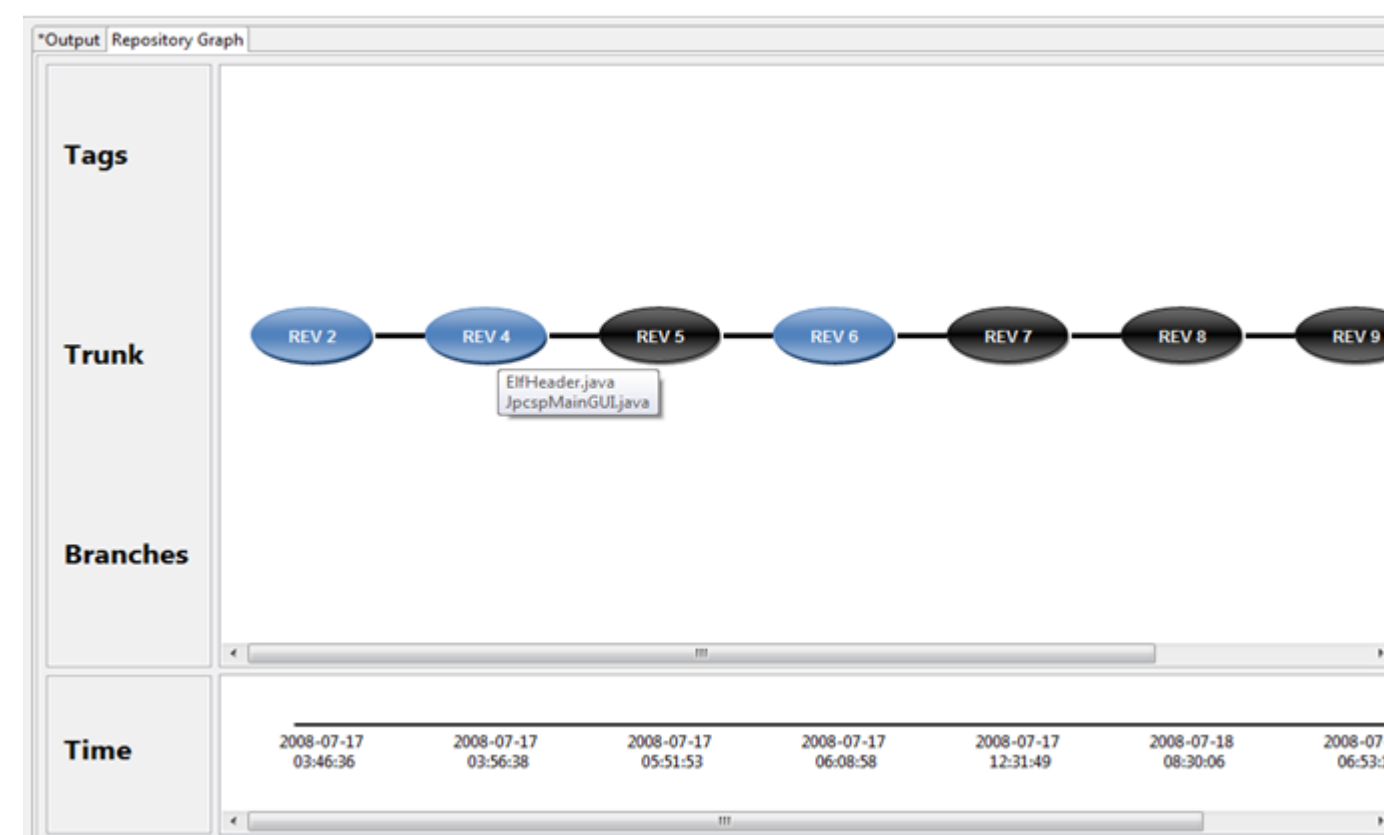
## IMPLEMENTATION



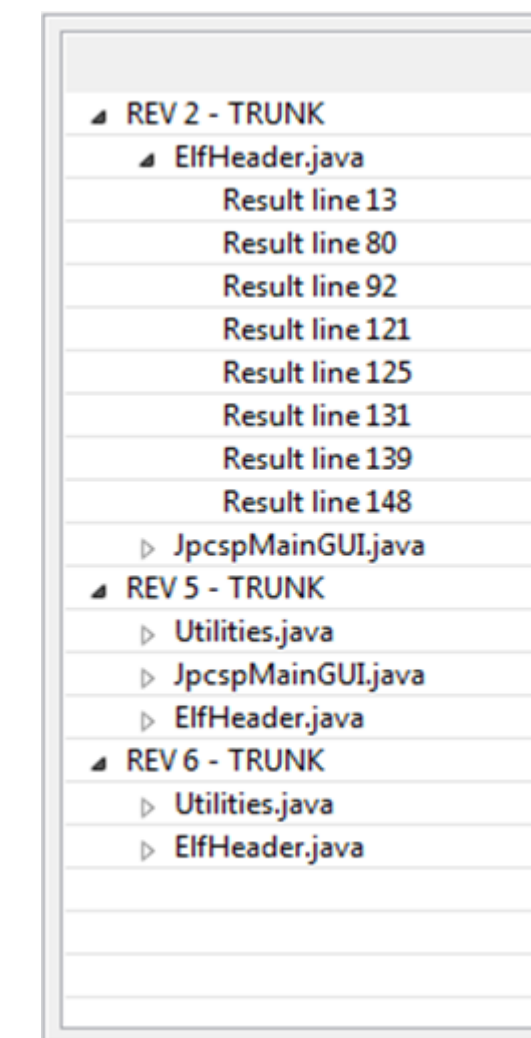
Choose a Repository & Revision Range



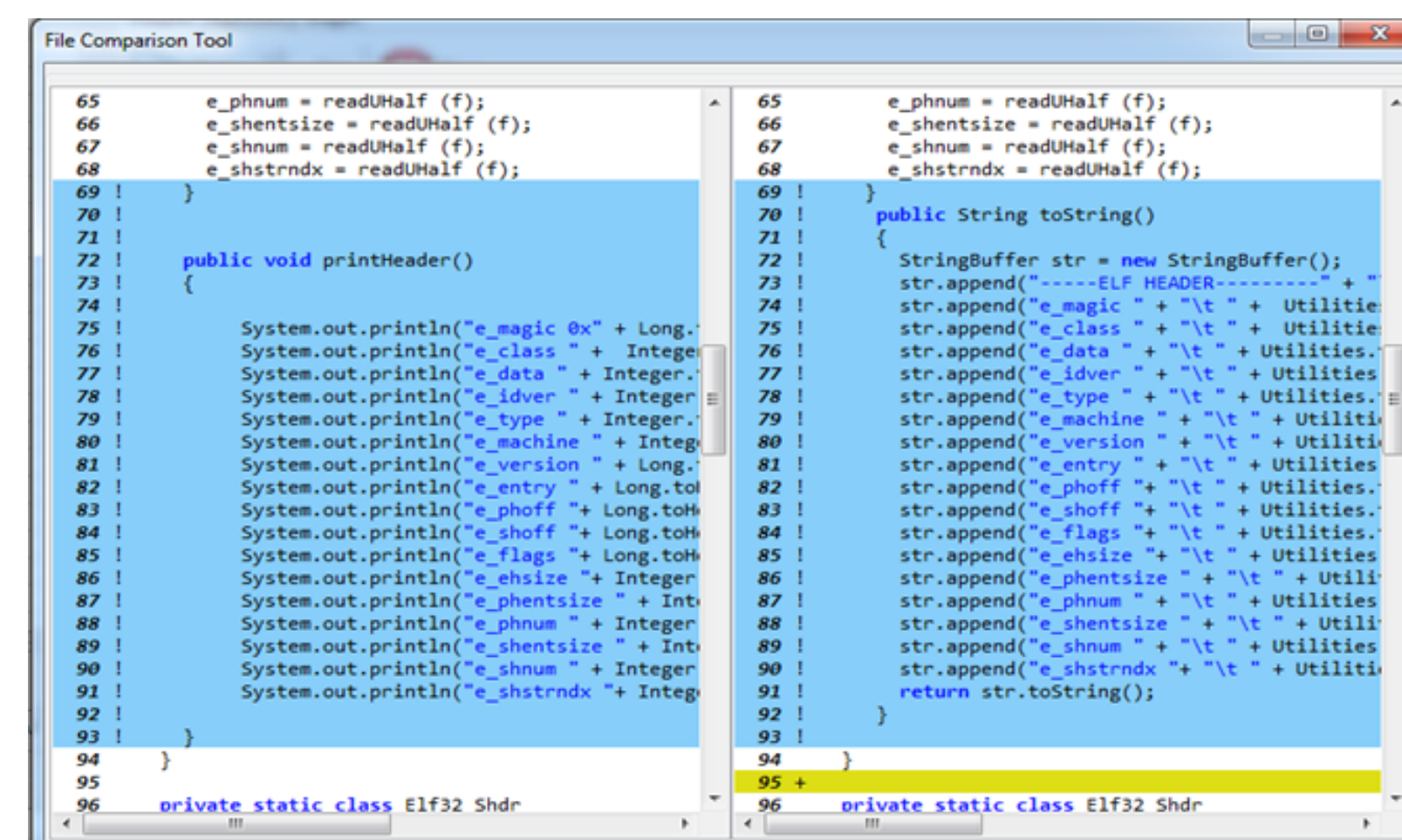
Execute a Query



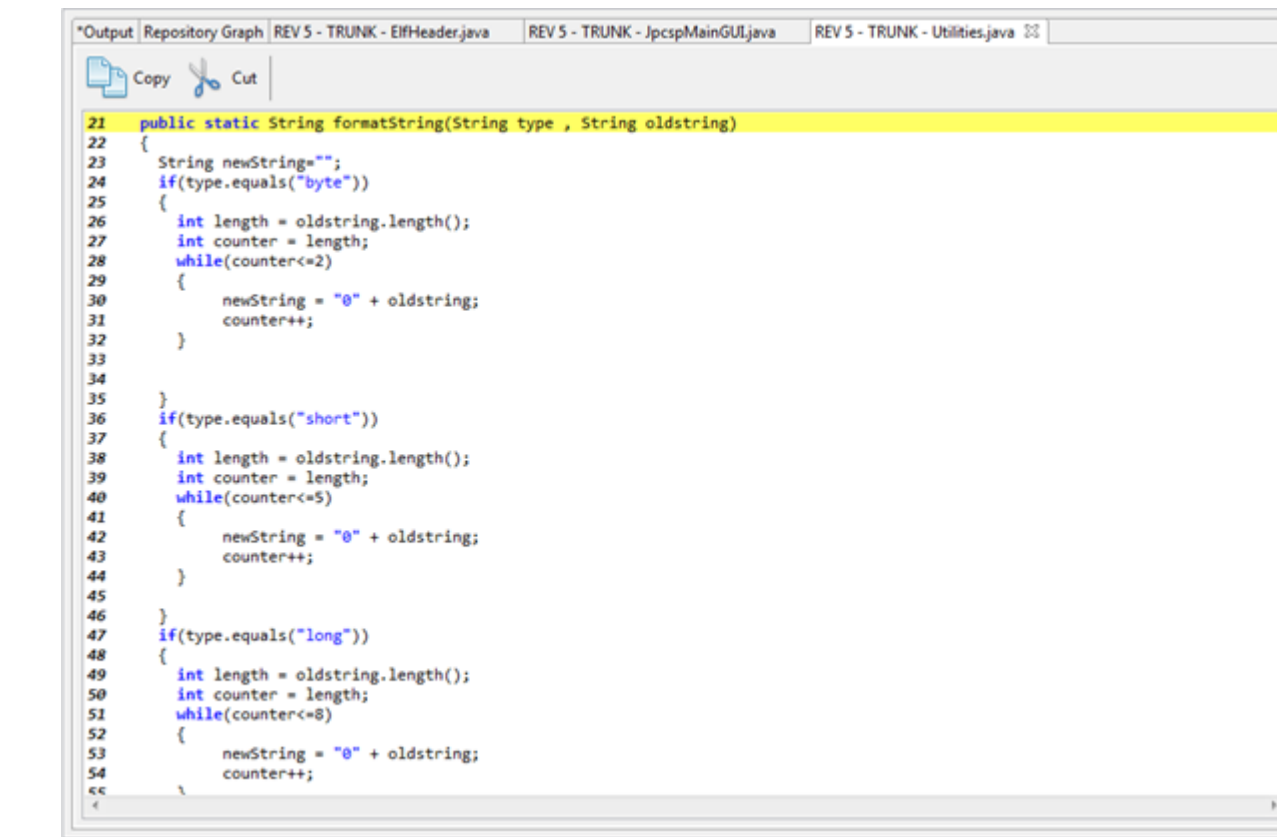
Evaluate results over time in repository



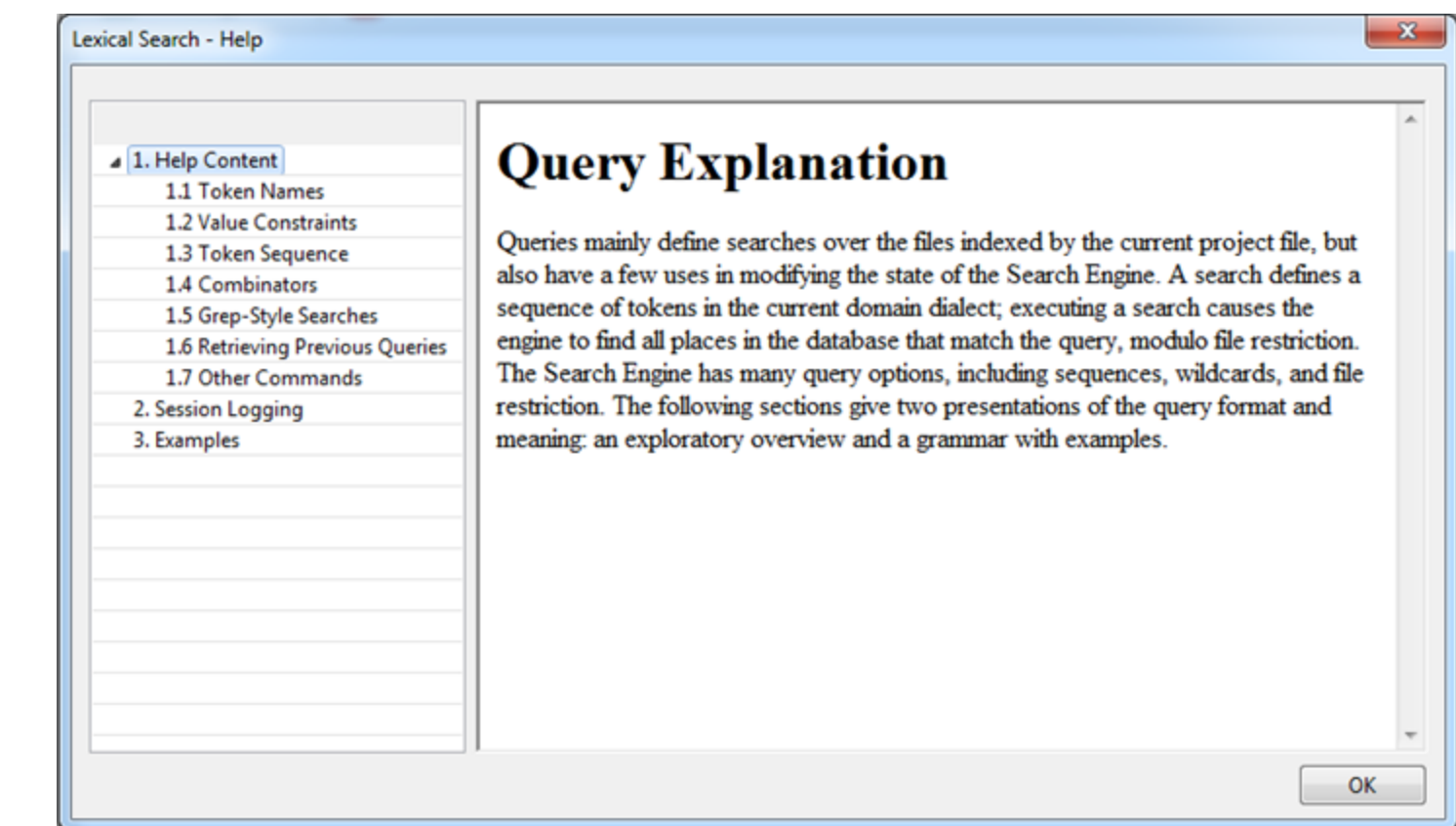
Evaluate individual files



Compare file differences across revisions

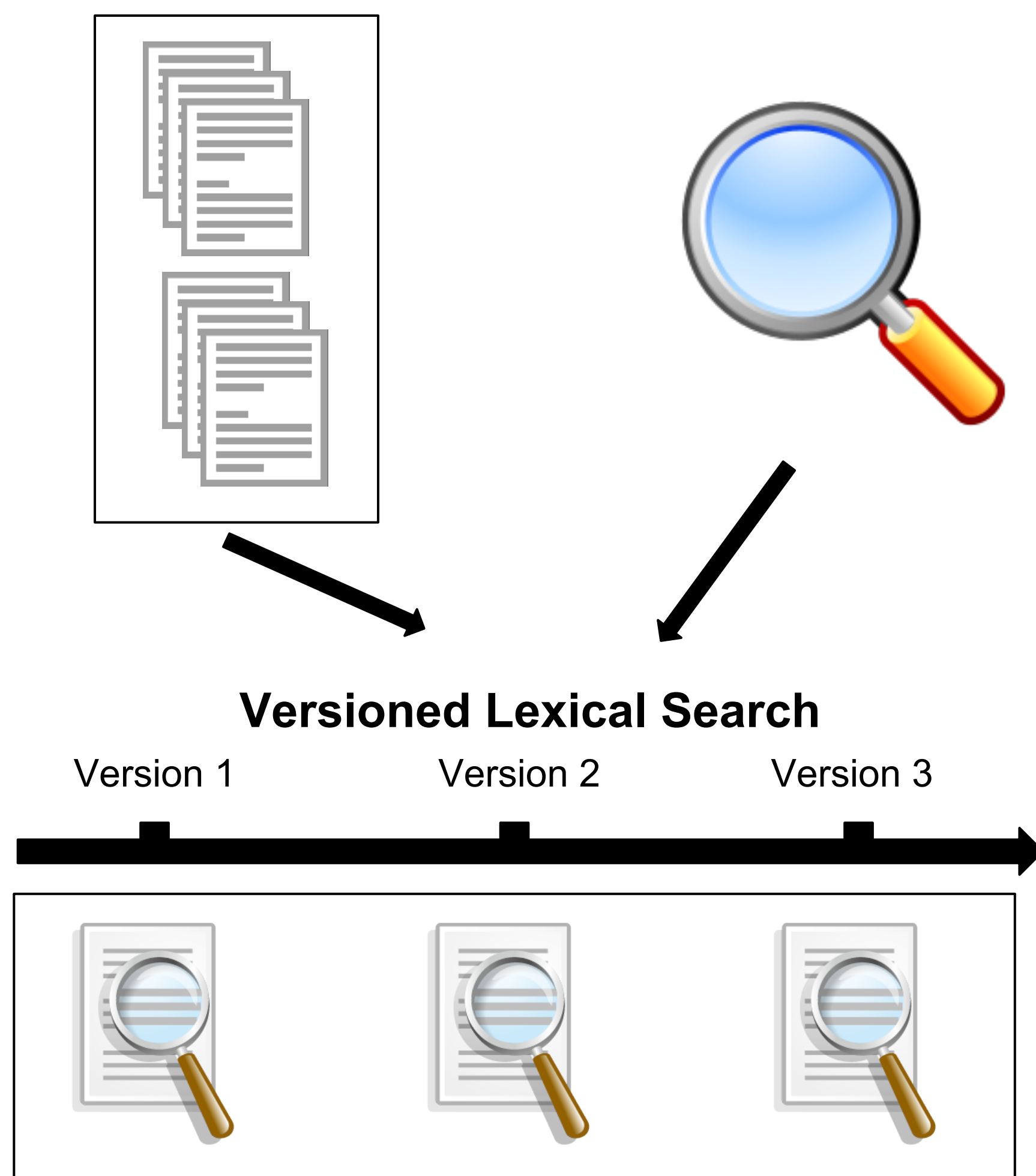


File Editing



Integrated Help

## Version Control Lexical Search



## APPROACH

- Semantic Designs (SD) builds a variety of software development tools. One of them is a lexical search system for source code. Utilizing this toolset provides half of the foundation necessary to build our solution upon.
- Apache Subversion (SVN) is a popular, open-source, revision control system that acts as the second half of the foundation for our solution.
- Tying these components together is a software back-end layer that facilitates communication between the user and the above tools by performing the necessary translations and interpretations necessary for each
- A Graphical User Interface (GUI) for the user to perform searches and evaluate results.

## CONCLUSION

We have described and implemented an approach that is capable of performing a lexical search across a version repository that yields meaningful results beyond simple textual matches. This system enables the ability to see the evolution of a result over time and its relationship to other components in the larger system.

## FUTURE WORK

**Performance:** As this proof of concept leverages a third party tool which was not designed for this purpose, the performance of this application leaves much to be desired. Better integration into the utilized tools is suspected to yield positive results.

**Wider Support:** Currently only the Subversion Revision Control system is supported. Additional systems such as Git and Mercurial among others could be supported.

**IDE Integration:** Being a standalone tool, this proof of concept does not lend itself to the level of convenience a developer might require. Integrating this system into one or more popular IDEs would promote this tool's utility and potentially lower the entry barrier to use.

## REFERENCES

- Alex Garcia. *Versioned Lexical Search*. University of Wisconsin-Milwaukee, 2012.
- "Apache™ Subversion®" *Apache Subversion*. 09 April 2013. Web. 09 Apr. 2013 <<http://subversion.apache.org/>>
- "Source Code Search Engine" *Semantic Designs*. 09 April 2013. Web. 09 Apr. 2013 <<http://www.semdesigns.com/Products/SearchEngine/>>