

|  |
| --- |
| **CSCI Department – CS Capstone I (P445) – Software Requirement Specification** |
| PHYLOGEN: PHYLOGENETICS DATA INTERFACE |
|  |
| **Kevin Florian | Joshua Rowe | Erica Wright** |
| Team “Armadillo Island Software” |

|  |
| --- |
|  |

## PURPOSE

This system’s purpose is to help the biology world by aiding in the creation of Nexus files for Phylogenetic analysis and to perform a subset of the Phylogenetic analyses based on the created Nexus file. This system will alleviate much of the time and functionality costs of the systems currently in place to perform Nexus file creation and Phylogenetic analysis because the current systems are extremely strenuous and time consuming, whereas the goal for this system is to create an easily used UI to aid the user in their file creation and basic analysis to allow students beginning in phylogeny to have the ability to participate in the analyses.

## DEFINITIONS

* *Taxa* - taxonomic category or group, such as a phylum, order, family, genus, or species (singular: *taxon*).
* *Phylogeny* - is the study of phylogenesis, or the evolutionary history, development and relationships among groups of organisms (e.g. species, or populations)
* *Morphological data* – Morphology, a branch of biology dealing with the study of the form/structure of organisms and their features

## *Genbank* - is the NIH (National Institute of Health) genetic *sequence* database – an annotated collection of all publicly available DNA/RNA sequences.

## *Nexus file* – as of now, the plain text file that PAUP, and systems like it, use to compare taxa’s morphological/sequence data to determine how closely a set of organisms are evolutionarily related.

## *Paup-* is a computational phylogenetics program for inferring evolutionary trees (phylogenies)

## SYSTEM OVERVIEW

This system will work by allowing students to enter the taxa they wish to analyze in a UI with text boxes, menus, and basic form controls. As they are entering all of the information they desire, the UI will perform validation tasks to ensure that the student is entering appropriate data. Once the user has entered data, they can create a Nexus file, which they can then view and use for our subset of analyses/features, or the user can take the file and apply it to other phylogenetic analysis software, such as PAUP, which is currently being utilized here at IUS. Potentially PHYLOGEN will have a small subset of tree visualization features, so the user may be able to generate a downloadable image, representing the relationships among taxa given. This image file would be exportable in JPEG, PNG, or PDF format to allow for universal viewing. An additional goal is to also allow this system to be mobile-friendly so that the user is able to prepare and analyze Nexus files with their phone.

## REFERENCES

*Dr. Taylor, David. Personal Interview (And Sponsor).Sept. 01, 2015. Sept. 11, 2015. Sept. 25, 2015.*

*Collins English Dictionary - Complete & Unabridged 10th Edition*. HarperCollins Publishers. 29 Sep. 2015. <Dictionary.com: <http://dictionary.reference.com/browse/taxa>>.

*“Phylogenetics”. Wikipedia, 2015 <<https://en.wikipedia.org/wiki/Phylogenetics>>*

*"phyloXML: XML for evolutionary biology and comparative genomics". Han M.V. and Zmasek C.M. (2009)* [*http://www.biomedcentral.com/1471-2105/10/356/*](http://www.biomedcentral.com/1471-2105/10/356/)

*PAUP (Phylogenetic Analysis Using Parsimony). David Swofford (1998). <http://paup.csit.fsu.edu/>*

## PRODUCT PERSPECTIVE

*System Interfaces*

Computational Phylogenetics is a data-dense application of analytical algorithms. The PHYLOGEN program will simplify the handing of this data by abstracting its structure through the use of XML – specifically *phyloXML*. In addition, current data standards in phylogenetic analysis are outdated and limited. The use of XML with a phylogenetic tree scheme modernizes this data, makes it easy to parse and transport, and efficiently contains the input from the improved user interface. This XML based system also lends itself to future improvements/additions.

*User Interfaces*

The current forms of user input with the NEXUS file format are overwhelmingly manual (sequence entry, characteristic code binding, structuring, etc.). PHYLOGEN will speed up the data collection process by automating much of the requirements in the NEXUS file format (syntax, markers, pre-process directives, etc.). It will also abstract the interaction of the data by giving the user simple controls, including buttons, drop-lists, radio buttons, form input boxes, among other options that will naturally arise in later design phases.

*Hardware interfaces*

The hardware interfaces PHYLOGEN will target are a number of different form factors: desktop & laptop computers, tablets, and smartphones. Though these may run on various Operating System platforms, the application will retain its functionality and usefulness no matter what device the user has. The junction of hardware to UI will be through keyboard and mouse, touch and/or pen, and perhaps eventually even voice.

*Software interfaces*

Because the main objective of this application will be the generation and manipulation of NEXUS files, primarily used by PAUP/McClade, PHYLOGEN will interact directly with these programs. Nexus files can come and go from either side of the interaction: PHYLOGEN-created files fed into PAUP for evolution tree analysis; existing analysis data from PAUP fed into PHYLOGEN for editing.

In addition, access to the genetic 0sequence database, GenBank, will be necessary for sequential data analysis. (API interface for custom data sets: http://www.ncbi.nlm.nih.gov/home/tools.shtml)

*Communication Interfaces*

Because the program may need to download genetic sequence data, the user’s device must have a connection to the internet. Any user interaction with this data extraction will be abstracted as much as possible. The GenBank API will take care of anything beyond the user selecting data samples.

Bluetooth connectivity, a USB port, or other such hardware/software portals integral to the device OS can be used to transport data to/from the device, and on to PHYLOGEN.

*Memory Constraints*

The data PHYLOGEN will be handling as a data interface for PAUP can be large, but is typically not massive. Sequential Genetic data is very long by nature, but its overall effect on memory is minimal due to its text form. PHYLOGEN will be handling only one file at a time – albeit with many data entries – this will be akin to the memory load of a text editor.

*Operations*

A list of operations PHYLOGEN might perform, per the requirements elicitation, include: file import, file creation, file merge, file export, edit data (text), input data (text), edit tags/markers, edit header info, comments, colorize, graphical data visualization, error check data, and other such operations that will be detailed into later design phases.

## PRODUCT FUNCTIONS

The functions of this product will be primarily to create, view, and edit Nexus files, and to offer a subset of analyses and features with a given Nexus file. These peripheral/secondary features will supplement the students’ use of PAUP where the complex algorithm analysis of the Nexus data occurs.

## USER CHARACTERISTICS

The primary users that will be utilizing PHYLOGEN are students learning the basics of Phylogenetics. The goal is for students at any campus to utilize this system, but for now, only those here at IUS will be using it. Dr. David Taylor, Biology professor here at IUS, suggested that even Biologists/researchers at his level might utilize the Nexus file generating aspect of this system – a testament to how strenuous the current Nexus file generation protocol is.

## CONSTRAINTS, ASSUMPTIONS, AND DEPENDENCIES

As of now, the biggest constraint is determining how to generalize the vast amount of taxa into some enumerable datatype to allow for applicable analysis. In addition to allowing morphological data entry, Dr. Taylor suggested creating this system around taxa sequence data banks, such as GenBank. One large assumption with this aspect is usability and easy integration of sequential data through the GenBank API. Utilizing a 3rd party dependency allows for some uncertainty in the back end of our system, however its benefit and expanded functionality makes PHYLOGEN that much more promising. As for the internal data control, PHYLOGEN will depend on phyloXML – a free, open-source XML language/schema available to the phylogenetics field of bioinformatics.

## External Interface Requirements

The program has to have a GUI interface that allows users to enter information pertaining to the collection and analysis of varying taxa. The UI will contain several control types for users to enter such data that will eventually be used to generate a nexus file. Fonts and layout must adhere to a modern and easy-to-read design language, the UX (user experience) flow must be intuitive, and various menus and shortcut keys will be used for the various data entry functions.

The hardware interface will be entirely through the Win32 API that is included with Visual Studio. This will be the case initially, but if we port do a different platform, then we will adapt the product to take advantage of that platform’s system calls. This means that any computer that can run Windows will be able to run our software. In fact, the new Windows universal development platform (UWP) will enable us to port to other platforms as well.

## Functional Requirements

Our advisor had stated that our requirements are mostly based on what students will use to get from data collected to the generation of a NEXUS file. This includes the users being able to enter Taxon information (morphological/sequence data) and manipulate that data in various ways (editing, colorizing/visualizing, exporting). After some data has been entered, the user will be able to access previously entered information in the form of drop down menus for easy access in continuing to enter information. Other requirements include the ability for the program to correctly generate a NEXUS file that can then be input into a phylogenetic analysis program. Lastly, if time permits, an additional requirement could be that the program processes nexus files so users can graphically manipulate evolutionary trees to see if their guesses are correct when the analysis is completed.

## Performance Requirements

The program is not required to do major computation at this point, so it just has to be responsive to the user who is inputting data into it. The output of the actual NEXUS file should be quick, but it doesn’t have to be optimized as long as it outputs the nexus file in a reasonable amount of time. The program performs no data-heavy analysis – its operations are mainly text entry, option selection, and real time user-adjustments.

## Design Constraints

There are only a few design constraints. Firstly, the software should be developed to run on Windows since that is primarily what the University uses. This is the biggest constraint as it affects the very tools we use to develop with. This will involve familiarization with XAML and utilize the C# language and .NET framework. In terms of UI, the design of on-screen elements must necessarily be intuitive, easy to understand at an undergraduate-student level, and efficient in terms of on-screen layout/placement and rendering. Other than minor platform specifics in the future, there are no other design constraints that limit the project.

## Logical Database Requirements

Our project does not require the use of a database with its current scope. Through the use of XML data structures based on phyloXML, our software will almost solely process and run on files (XML and Nexus). We may use a lite-version database, however, to standardize the input and output of data and other non file-based elements.

## SOFTWARE SYSTEM ATTRIBUTES

Our software should be reliable in the sense that it does not crash, is responsive to user requests, and creates accurate NEXUS files that will be used for the analysis portion of the work users are doing.

Our software should be available in a current, up-to-date form for all users who wish to use it. This takes into account the ability to package the software for remote deployment on this campus.

Our software does need to be secure, protecting the data integrity of information entered into the system (oftentimes very lengthy and extremely specific) and the computer system in which it is being ran on. It must have an auto-save feature so that in the unlikely event that the app/device crashes, the user can recover any lengthy/important edits made to the data.

* Introduction
  + Purpose ***Erica***
  + Definitions ***Erica***
  + System overview ***Erica***
  + References ***Erica***
* Overall description
  + Product perspective
    - System Interfaces ***Kevin***
    - User Interfaces ***Kevin***
    - Hardware interfaces ***Kevin***
    - Software interfaces ***Kevin***
    - Communication Interfaces ***Kevin***
    - Memory Constraints ***Kevin***
    - Operations ***Kevin***
    - Site Adaptation Requirements ***Kevin***
  + Product functions ***Erica***
  + User characteristics ***Erica***
  + Constraints, assumptions and dependencies ***Erica***
* Specific requirements
  + External interface requirements ***Joshua***
  + Functional requirements ***Joshua***
  + Performance requirements ***Joshua***
  + Design constraints ***Joshua***
  + Logical database requirement ***Joshua***
  + Software System attributes ***Joshua***
  + Other requirements ***Joshua***