**The Skyentists**

**Final Binder**

**CSCI 426**

**Fall 2019**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | The Skyentists | |  |  |
| Contact INFORMATION | | | | |
| Company name: | | Numerical Terradynamic Simulation Group | | |
| CEO/Primary Stakeholders | | Arthur Endsley | | |
| Email addresses | | [endsley@umich.edu](mailto:endsley@umich.edu) | | |
| Software Developer names | | Max Thibeau, Jake Pennington, Lucas Hamilton, Mark Matas | | |
| Email addresses | | [Max.thibeau@umontana.edu](mailto:Max.thibeau@umontana.edu), [jake.pennington@umontana.edu](mailto:jake.pennington@umontana.edu), [lucas.hamilton@umontana.edu](mailto:lucas.hamilton@umontana.edu), [mark.matas@umontana.edu](mailto:mark.matas@umontana.edu) | | |

|  |
| --- |
| CHECKLIST (make sure the following is included in your final binder) |

|  |  |
| --- | --- |
|  | Cover label |
|  | Copy of this checklist, completed |
|  | Dividers |
|  | Table of Contents |
|  | Student Rights forms |
|  | List of edits to deliverables |
|  | Team Charter V1 |
|  | Team Charter Final Version |
|  | Project Charter V1 |
|  | Project Charter Final Version |
|  | Requirements Specification V1 |
|  | Requirements Specification Final Version |
|  | Systems Modeling spec V1 |
|  | Systems Modeling spec Final Version |
|  | Low fidelity prototype V1 |
|  | Low fidelity prototype Final Version |
|  | User Feedback specification V1 |
|  | User Feedback specification Final Version |
|  | Implementation Plan |
|  | Prototype 2 plus User Feedback doc |
|  | Final Presentation – slide deck (slides can be shrunk so multiple fit per printed page) |
|  | Wk 5 -- Progress Report |
|  | Wk 8 -- Progress Report |
|  | Wk 13 -- Progress report |
|  | Signatures – All deliverables and forms are signed by all required stakeholders |
|  | Miscellaneous items (notes from client meetings, gantt chart, etc.) |

**Table of Contents**

Intellectual Property Agreements – 5

Document Changes – 9

Old Team Charter – 10

Final Team Charter – 15

Old Project Charter – 21

Final Project Charter – 27

Old Requirements Specification Document – 34

Final Requirements Specification Document – 42

Old Systems Modeling Document – 52

Final Systems Modeling Document – 58

Old Low-fidelity Prototypes Document – 64

Final Low-fidelity Prototypes Document – 80

Old User Feedback for UI Prototype – 96

Final User Feedback for UI Prorotype – 98

Implementation Plan – 100

High Fidelity UI Prototype – 102

Final Presentation – 108

Progress Report 1 – 111

Progress Report 2 – 112

Progress Report 3 – 113

Team Gantt – 114

**Intellectual Property Agreement**

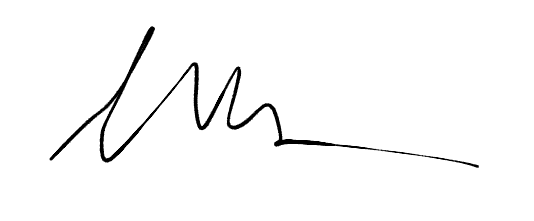
**Between**

**Max Thibeau and Arthur Endsley**

I hereby assign to Arthur Endsley all my right, title and interest in any deliverables developed under the scope of Earth System Models from Flux Tower Data to be completed as part of my practicum assignment in CSCI 426 at the University of Montana for the Fall 19 and Spring 20 semesters. Furthermore, I will disclose to the Arthur Endsley and the Instructor of CSCI 426 all potentially patentable inventions related to the Earth System Models from Flux Tower Data so that the question of whether the Arthur Endsley has rights to the invention can be determined with the assistance of the University’s research office.

I hereby agree that I am now under no consulting or other obligations to any third person, organization or corporation in respect to rights in inventions which are, or could be reasonably construed to be, in conflict with this agreement. I will not enter into any agreement creating patent obligations in conflict with this agreement.

This agreement is effective on the date of signature below.

****

Signature:

Printed Name: Max Thibeau

Date: 09/06/19

UM ID: 790738727

**Intellectual Property Agreement**

**Between**

**Mark Matas and Arthur Endsley**

I hereby assign to Arthur Endsley all my right, title and interest in any deliverables developed under the scope of Earth System Models from Flux Tower Data to be completed as part of my practicum assignment in CSCI 426 at the University of Montana for the Fall 19 and Spring 20 semesters. Furthermore, I will disclose to the Arthur Endsley and the Instructor of CSCI 426 all potentially patentable inventions related to the Earth System Models from Flux Tower Data so that the question of whether the Arthur Endsley has rights to the invention can be determined with the assistance of the University’s research office.

I hereby agree that I am now under no consulting or other obligations to any third person, organization or corporation in respect to rights in inventions which are, or could be reasonably construed to be, in conflict with this agreement. I will not enter into any agreement creating patent obligations in conflict with this agreement.

This agreement is effective on the date of signature below.



Signature:

Printed Name: Mark Matas

Date: 09/06/19

UM ID: 790743477

**Intellectual Property Agreement**

**Between**

**Jake Pennington and Arthut Endsley**

I hereby assign to Arthur Endsley all my right, title and interest in any deliverables developed under the scope of Earth System Models from Flux Tower Data to be completed as part of my practicum assignment in CSCI 426 at the University of Montana for the Fall 19 and Spring 20 semesters. Furthermore, I will disclose to the Arthur Endsley and the Instructor of CSCI 426 all potentially patentable inventions related to the Earth System Models from Flux Tower Data so that the question of whether the Arthur Endsley has rights to the invention can be determined with the assistance of the University’s research office.

I hereby agree that I am now under no consulting or other obligations to any third person, organization or corporation in respect to rights in inventions which are, or could be reasonably construed to be, in conflict with this agreement. I will not enter into any agreement creating patent obligations in conflict with this agreement.

This agreement is effective on the date of signature below.

Signature:



Printed Name: Jake Pennington

Date: 09/06/19

UM ID:

**Intellectual Property Agreement**

**Between**

**Lucas Hamilton and Arthur Endsley**

I hereby assign to Arthur Endsley all my right, title and interest in any deliverables developed under the scope of Earth System Models from Flux Tower Data to be completed as part of my practicum assignment in CSCI 426 at the University of Montana for the Fall 19 and Spring 20 semesters. Furthermore, I will disclose to the Arthur Endsley and the Instructor of CSCI 426 all potentially patentable inventions related to the Earth System Models from Flux Tower Data so that the question of whether the Arthur Endsley has rights to the invention can be determined with the assistance of the University’s research office.

I hereby agree that I am now under no consulting or other obligations to any third person, organization or corporation in respect to rights in inventions which are, or could be reasonably construed to be, in conflict with this agreement. I will not enter into any agreement creating patent obligations in conflict with this agreement.

This agreement is effective on the date of signature below.

Signature:

Printed Name: Lucas Hamilton

Date: 09/06/19

UM ID:

**Document Changes**

1. Team Charter
   * Initially created on 9/12/19
   * Edits after Professor Reimer’s feedback on 9/20/19
     + Edited Lucas’ bio to be less awkwardly written
     + Fixed wording of beginning sections
     + Removed instructions left in the sections
2. Project Charter
   * Initially created on 9/19/19
   * Edits after viewing Professor Reimer’s feedback on 9/27/19
     + Altered the stakeholder section to have proper wording (replacing “Arthur” with “Client”) and re-worded to make more clear
     + Added introductory paragraph to the measurables section
3. Requirement Specification
   * Initially created on 10/7/19
   * Edits after viewing Professor Reimer’s feedback on 10/14/19
     + Gave context to the questions asked to the users in User Feedback section
     + Provided more in-depth explanation for some of the functional requirements
     + Added better phrasing and some examples to some of the non-functional requirements
     + All team member signatures were added
   * Added medium priority requirements based on client feedback on 11/2/19
   * Edits after Professor Reimer’s feedback on 11/9/19
     + Moved the changes list from below the signatures to above the first section
     + Added the updated date to the cover page
     + Fixed some left-over grammatical errors
4. Systems Modeling
   * Initially created on 10/14/19
   * Edits from Professor Reimer’s feedback on 10/29/19
     + Edited sentences to improve clarity in the Overview section
     + Included a brief sentence explaining a high-level overview of what the calibration process will accomplish
     + Changed the diagram 4 description to reflect the RECO optimization
5. UI Prototyping
   * Initially created on 10/21/19
   * Professor Reimer’s feedback on 10/28/19 (mostly in-class) for Medium-High Fidelity UI Prototype
6. User Feedback
   * Initially created on 10/28/19
   * Edits after Professor Reimer’s feedback 11/12/19
     + Inputted statement of acknowledgement to the signatures
     + Added headings to each section
     + Created an Overview section to describe why feedback is important
     + Added how the feedback list is being categorized

Old Team Charter

*“A team charter is a document that is developed in a group setting that clarifies team direction while establishing boundaries. It is developed early during the forming of the team. The charter should be developed in a group session to encourage understanding and buy-in.*

*The team charter has two purposes. First, it serves as a source for the team members to illustrate the focus and direction of the team. Second, it educates others (for example the organizational leaders and other work groups), illustrating the direction of the team.”*

<https://www.lce.com/Team-Charters-What-are-they-and-whats-their-purpose-1219.html>

Prepared by: The Skyentists

Date: 09/06/19

1. All project deliverables and sections with deliverables should be polished, professional, proof-read, well-labeled, and in order.
2. Keep the headings below in your deliverable, but remove the descriptive text.
3. When you are listing specific items/element, please use bullet points. When you are describing or adding detail, use paragraph prose.
4. All project deliverables that are group based should be submitted to Moodle by the Project Manager only.

# SKYentists

We are the SKYentists, a driven software development team determined to deliver a quality product.

# Goals

Our goal as a team is to work as a cohesive unit to build a project that we can take pride in. We’ll learn how to take advantage of each other’s strengths and make up for our weaknesses. Our goal is also to ensure that the client feels well-respected and is an integral part of the project. Another goal of our team is to expand our knowledge in areas that we do not fully understand and incorporate this into the project or further in our careers.

# Members

Jake Pennington, 1-406-459-5710,

[jake.pennington@umontana.edu](mailto:jake.pennington@umontana.edu)

Jake grew up in the small town of Elliston, Montana where he enjoyed mathematics and technology. He discovered his passion for programming at the age of 14 and has continued that lifestyle to this very day. Jake attended the University of Montana where he was awarded a B.A. in Mathematics. He is currently a graduate student at UM working on an M.A. in Mathematics while also completing a B.S. in Computer Science. He has worked for Dr. Oliver Serang where he tested and debugged software, optimized parameters, and developed algorithms for bioinformatics.

Jake has a lot of experience in object-oriented programming, but also has a healthy exposure to functional and component-based models. He can do it all, but especially enjoys back-end development, algorithm design, and technical advising. His responsibilities for this project will likely include programming the back-end, ensuring code meets functional and non-functional requirements, providing adequate documentation and testing for the codebase, and ensuring license compatibility will all external libraries.

Max Thibeau, 1-406-214-1680,

[max.thibeau@umontana.edu](mailto:max.thibeau@umontana.edu)



Max Thibeau was born and raised right here in Missoula, Montana. Max always enjoyed solving puzzles, and naturally found programming to be a good outlet for problem solving. Max has gone through most of the CS program and has had some valuable work experience under Dr. Oliver Serang. He’s currently helping the lab develop an expectation maximization algorithm for solving protein inference problems. This is part of an in house Python protein inference library he helped write that’s heavily Object Oriented. The foundation of this library is a programming language Max wrote that generates CPP code for Evergreen Forest, a graphical models solver.

Given this experience, Max is willing to help with anything but will be most interested in being the team architect for the project. Getting a grand scope of the entire process from start to finish and making objects to abstract as much of the process as possible would be his goal. If the interface portion of the process is finished, Max would also be interested in learning the process of satellite calibration with ground truth data to see if there’s some way to automate the process.

Lucas Hamilton, 1-406-290-4426, [lucas1.hamilton@umconnect.umt.edu](mailto:lucas1.hamilton@umconnect.umt.edu)

Lucas Hamilton was raised partially in Newbury, Massachusetts, where he would later move to Montana to live in the town of Reed Point, Montana, to this current day. Lucas started coding in high school as a part of the Code Montana program, where he gained an interest in programming. Lucas has gained gain most of his experience through his course work with the University of Montana, where he learned many languages that consist of JavaScript, Java, HTML/CSS, Python, C/C++, MySQL, Oracle SQL, and R.

Lucas’s Primary Role would mostly consist of helping with the software development of the project, with a close secondary role as the Data Analyst if needed due to his experience in Data Visualization in the UM course of the same name. Lucas will, however, also help with any role that needs to be filled along with the aiding of fixing problems that may arise in other roles. The main goal that Lucas wants to get out of this project would be to learn more about how the development cycle works and to gain experience for the working world.

Mark Matas, 1-480-318-0106,

[mark.matas@umontana.edu](mailto:mark.matas@umontana.edu)

Mark Matas, both an Illinois and Arizona native, has been around Missoula, Montana for about six years now due to having family in town. Mark has a B.S. in Biology with a minor in Mathematics and currently is in his last year of earning his B.S. in Computer Science. He gained a lot of useful skills in his time at the University of Montana, from not knowing how to program to becoming proficient in software development. Mark has programming experience in Python, C/C++, Java, JavaScript, Ruby/Ruby on Rails, and SQL. He has worked the past two summers as an Implementation Consultant for Fast Enterprises and has been working for Dr. Rob Smith on campus for three years.

Mark has always enjoyed a creative outlet, from hiking up mountains to drawing cartoons to problem-solving during programming and gaming. Mark has been elected the project manager for the SKYentists, so he is willing to do what needs to be done to deliver a working, well-documented program based on client requirements to ensure the upmost satisfaction. He is interested in how the process of satellite calibration works, as well as seeing how his Biology degree will help make informed decisions about the project.

# Values & Ethics

* Avoid harm: do not hurt anyone, do not make Arthur OR Yolanda angry (ACM)
* Open Communication: always be professional, let other members know if you are going to absent/late, members should communicate when progress is made and when they need help
* Shared Responsibility: share the workload evenly, hold one another accountable for an even load, if one member fails then we all fail
* Initiative: take charge and have the want to complete the project
* Punctuality: keep things moving, show up on time when needed, arrive prepared
* Attendance: always show up and communicate when not able to attend

# Rules & Expectations

* Meetings: set up my Mark in Gantt, everyone needs to be there unless they communicate to everyone their absence, everyone’s opinion will be heard in meetings. There will always be a designated notetaker.
* Client: Mark provides weekly status updates to the client, CC’ing other members of the group when contacting with client, always speak, write and act professionally. The client’s time will be valued highly.
* Team Decision: We will listen to everyone’s opinions, we’ll use cost benefit analyses and value identification to frame our opinions, and arrive at a majority decision (unanimous decision preferred).
* Communication: open and professional whether it be in meeting or in email, communication between members done through text, texting encouraged to be casual.
* Documents: created and reviewed/edited by everyone, submitted by anyone, shared through email with every team members. Max will maintain the team binder.

# Schedule

* Class Meeting Times: Monday and Wednesday at 3-4:20 in SS 362
* Weekly Team Meetings: held on Fridays at 3-4 to talk about the work done during the week and sending the weekly status update to the client.
* Hour will be logged into Gantt by Mark
* Potential for more meetings
* Class Client Meeting Dates: 9/16, 10/7, 10/28, and 11/25 (all on Mondays)
* First Scheduled Client Meeting: 9/16 during regular scheduled class time
* Progress Reports: Week of 9/23, Week of 10/14, and Week of 11/18
* Client Availability: Flexible schedule, works on UM campus
* Member Availability
* Mark: Tuesday-Friday any time after 3, Saturday-Monday open availability
* Jake: Monday and Wednesday after 4, Thursday 12:30-3, Friday after 3, Saturday-Sunday open availability
* Max: Monday-Friday 1-5, Saturday-Sunday open availability
* Lucas: Monday and Wednesday 11-3, Tuesday after 3, Thursday after 1, Friday-Sunday open availability

# Conflict Resolution Plan

Having a conflict resolution plan in place is important because conflict can make it hard for people to think clearly about an issue. Having a predetermined series of steps to follow makes it easy to take personal bias out of a conflict and look at the issues objectively.

While it would be nice to claim there is no possibility of conflict, this simply does not represent reality. Anytime a group is working on a project, people are bound to have different ideas about how to do things. Most of the time the group can reach an agreement about how to move forward but there are times when resolving these issues is not so simple and conflict can arise. Because conflict, to some degree, is all but inevitable; it is prudent to have measures in place to resolve such conflict. Our process for technical/developmental conflict resolution is described below:

Conflict Resolution Plan (for both technical and development conflicts):

1. Only the members in conflict will meet
   * Both sides give a cost benefit analysis of their side
   * Both sides discuss the values their stance holds
   * If no unanimous opinion is reached move to next step
2. Meet as a team
   * Both sides give a cost benefit analysis of their side
   * Both sides discuss the values their stance holds
   * Everyone discusses the issue at hand
   * A vote is held, the majority wins
   * If the vote ties move on to the next step
3. Bring conflict to Yolanda
   * Both sides will present their cost benefit analysis and values
   * Yolanda will give feedback to the team
   * The team votes once more, majority wins
   * If no majority is made Yolanda resolves conflict with absolute authority

# Execution and Acknowledgement

The team members hereby indicate by their signatures below that they have read and agree with the specifications of this charter.



Name & date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_12/12/19\_\_

Name & date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_12/12/19\_\_



Name & date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_9/06/19\_\_

Name & date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_9/06/19\_\_

Client Name & date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

# Final Team Charter

*“A team charter is a document that is developed in a group setting that clarifies team direction while establishing boundaries. It is developed early during the forming of the team. The charter should be developed in a group session to encourage understanding and buy-in.*

*The team charter has two purposes. First, it serves as a source for the team members to illustrate the focus and direction of the team. Second, it educates others (for example the organizational leaders and other work groups), illustrating the direction of the team.”*

<https://www.lce.com/Team-Charters-What-are-they-and-whats-their-purpose-1219.html>

Prepared by: The Skyentists

Date: 09/11/19

# SKYentists

We are the SKYentists, a driven software development team determined to deliver a quality product.

# Goals

Our goal as a team is to work as a cohesive unit to build a project that we can take pride in. We’ll learn how to take advantage of each other’s strengths and make up for our weaknesses. Our goal is also to ensure that the client feels well-respected and is an integral part of the project. Another goal of our team is to expand our knowledge in areas that we do not fully understand and incorporate this into the project or further in our careers.

# Members

Jake Pennington, 1-406-459-5710,

[jake.pennington@umontana.edu](mailto:jake.pennington@umontana.edu#_blank)

Jake grew up in the small town of Elliston, Montana where he enjoyed mathematics and technology. He discovered his passion for programming at the age of 14 and has continued that lifestyle to this very day. Jake attended the University of Montana where he was awarded a B.A. in Mathematics. He is currently a graduate student at UM working on an M.A. in Mathematics while also completing a B.S. in Computer Science. He has worked for Dr. Oliver Serang where he tested and debugged software, optimized parameters, and developed algorithms for bioinformatics.

Jake has a lot of experience in object-oriented programming, but also has a healthy exposure to functional and component-based models. He can do it all, but especially enjoys back-end development, algorithm design, and technical advising. His responsibilities for this project will likely include programming the back-end, ensuring code meets functional and non-functional requirements, providing adequate documentation and testing for the codebase, and ensuring license compatibility will all external libraries.

Max Thibeau, 1-406-214-1680,

[max.thibeau@umontana.edu](mailto:max.thibeau@umontana.edu)



Max Thibeau was born and raised right here in Missoula, Montana. Max always enjoyed solving puzzles, and naturally found programming to be a good outlet for problem solving. Max has gone through most of the CS program and has had some valuable work experience under Dr. Oliver Serang. He’s currently helping the lab develop an expectation maximization algorithm for solving protein inference problems. This is part of an in house Python protein inference library he helped write that’s heavily Object Oriented. The foundation of this library is a programming language Max wrote that generates CPP code for Evergreen Forest, a graphical models solver.

Given this experience, Max is willing to help with anything but will be most interested in being the team architect for the project. Getting a grand scope of the entire process from start to finish and making objects to abstract as much of the process as possible would be his goal. If the interface portion of the process is finished, Max would also be interested in learning the process of satellite calibration with ground truth data to see if there’s some way to automate the process.

Lucas Hamilton, 1-406-290-4426, [lucas1.hamilton@umconnect.umt.edu](mailto:lucas1.hamilton@umconnect.umt.edu)

Lucas Hamilton was originally from Newbury, Massachusetts, where he would later be raised in Reed Point, Montana. Lucas initially developed in interest in Computer Science during high school where he participated in the Code Montana program and really enjoyed the problem-solving portion of the program. Due to this initial spark, Lucas would go on to achieve the majority of his experience through his course work with the University of Montana. The classes Lucas had taken gave him experience in JavaScript, Java, HTML/CSS, Python, C/C++, MySQL, Oracle SQL, and R.

Lucas’s Primary Role would mostly consist of helping with the software development of the project, with a close secondary role as the Data Analyst if needed. Data analyst would be his second role due to his experience in Data Visualization, a course he took at the University of Montana. Lucas will, however, also help with any role that needs to be filled. Lucas will also help fix problems that may occur in other roles. The main goal that Lucas wants to get out of this project would be to learn more about how the development cycle works and to gain experience for the working world.

Mark Matas, 1-480-318-0106,

[mark.matas@umontana.edu](mailto:mark.matas@umontana.edu)

Mark Matas, both an Illinois and Arizona native, has been around Missoula, Montana for about six years now due to having family in town. Mark has a B.S. in Biology with a minor in Mathematics and currently is in his last year of earning his B.S. in Computer Science. He gained a lot of useful skills in his time at the University of Montana, from not knowing how to program to becoming proficient in software development. Mark has programming experience in Python, C/C++, Java, JavaScript, Ruby/Ruby on Rails, and SQL. He has worked the past two summers as an Implementation Consultant for Fast Enterprises and has been working for Dr. Rob Smith on campus for three years.

Mark has always enjoyed a creative outlet, from hiking up mountains to drawing cartoons to problem-solving during programming and gaming. Mark has been elected the project manager for the SKYentists, so he is willing to do what needs to be done to deliver a working, well-documented program based on client requirements to ensure the upmost satisfaction. He is interested in how the process of satellite calibration works, as well as seeing how his Biology degree will help make informed decisions about the project.

# Values & Ethics

The following of ethics is of upmost importance to the team to have the ability to determine the justifiable decisions and behaviors that affect the team and client interactions. The values will help the team with these decisions by informing the thoughts, words, and actions to guide how the team will act towards each other and in meetings.

* Avoid harm: do not hurt anyone, do not make the client or Yolanda angry
* Open Communication: always be professional, let other members know if you are going to absent/late, members should communicate when progress is made and when they need help
* Shared Responsibility: share the workload evenly, hold one another accountable for an even load, if one member fails then we all fail
* Initiative: take charge and have the want to complete the project
* Punctuality: keep things moving, show up on time when needed, arrive prepared
* Attendance: always show up and communicate when not able to attend

# Rules & Expectations

The following list includes the rules of the team to achieve high quality standards and ensure the completion of the project. Expectations are also included so that the team members and client know what to do and how to perform in order to create successful software.

* Meetings: set up by Mark in Gantt and communicated with the team members, everyone needs to be there unless they communicate to everyone their absence, everyone’s opinion will be heard in meetings, one team member will always be a designated notetaker
* Client: team leader provides weekly status updates to the client while CC’ing other members of the group, always speak and write and act professionally when communicating with the client, value the client’s free time
* Team Decision: every team member’s opinion will be heard and valued, use cost benefit analyses and value identification to frame our opinions to arrive at a majority decision (unanimous decision preferred)
* Communication: open and professional in meetings or email, member communication can also be done through casual texting
* Documents: created and reviewed by the group, edited by everyone, submitted by one team member, all updates shared through email with every team member, Max will maintain the team binder

# Schedule

* Class Meeting Times: Monday and Wednesday at 3-4:20 in SS 362
* Weekly Team Meetings: held on Fridays at 3-4 to talk about the work done during the week and sending the weekly status update to the client.
* Hours will be logged into Gantt by the team leader
* Potential for more meetings not at this set time
* Class Client Meeting Dates: 9/16, 10/7, 10/28, and 11/25 (all on Mondays)
* First Scheduled Client Meeting: 9/16 during regular scheduled class time
* Client Availability: Flexible schedule, works on UM campus
* Member Availability
* Mark: Tuesday-Friday any time after 3, Saturday-Monday open availability
* Jake: Monday and Wednesday after 4, Thursday 12:30-3, Friday after 3, Saturday-Sunday open availability
* Max: Monday-Friday 1-5, Saturday-Sunday open availability
* Lucas: Monday and Wednesday 11-3, Tuesday after 3, Thursday after 1, Friday-Sunday open availability
* Progress Reports: Weeks of 9/23, 10/14, and 11/18

# Conflict Resolution Plan

Having a conflict resolution plan in place is important because conflict can make it hard for people to think clearly about an issue. Having a predetermined series of steps to follow makes it easy to take personal bias out of a conflict and look at the issues objectively.

While it would be nice to claim there is no possibility of conflict, this simply does not represent reality. Anytime a group is working on a project, people are bound to have different ideas about how to do things. Most of the time the group can reach an agreement about how to move forward but there are times when resolving these issues is not so simple and conflict can arise. Because conflict, to some degree, is all but inevitable; it is prudent to have measures in place to resolve such conflict. Our process for technical/developmental conflict resolution is described below:

Conflict Resolution Plan (for both technical and development conflicts):

1. Only the members in conflict will meet
   * Both sides give a cost benefit analysis of their side
   * Both sides discuss the values their stance holds
   * If no unanimous opinion is reached move to next step
2. Meet as a team
   * Both sides give a cost benefit analysis of their side
   * Both sides discuss the values their stance holds
   * Everyone discusses the issue at hand
   * A vote is held, the majority wins
   * If the vote ties move on to the next step
3. Bring conflict to Yolanda
   * Both sides will present their cost benefit analysis and values
   * Yolanda will give feedback to the team
   * The team votes once more, majority wins
   * If no majority is made Yolanda resolves conflict with absolute authority

# Execution and Acknowledgement

The team members hereby indicate by their signatures below that they have read and agree with the specifications of this charter.



Name & date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_12/12/19\_\_

Name & date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_12/12/19\_\_



Name & date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_9/11/19\_\_

Name & date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_9/11/19\_\_

Client Name & date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

Old Project Charter

**Project Name:**

Earth System Models from Flux Tower Data

Prepared by:

The Skyentists

# Date:

9/19/19

# 1. Project Descrption & Problem

## 1.1 Background

This project involves creating a new calibration process for a climate change model that deals with different plant types and soil moisture. The model gets its test data from eddy flux towers located across the globe, as well as other scientific data. This improved calibration process will allow the user to tweak the model’s inputs and graph outputs to more accurately reflect the tower data. The sponsor of this project is the Numerical Terradynamic Simulation Group (NTSG), a group that researches tech development in certain Earth satellite data. The sponsor’s mission statement is "our primary mission is to develop capabilities to quantitatively describe the structure and function of ecosystems from regional to global scales using emerging technologies in satellite remote sensing, computational modeling and biophysical theory."

## 1.2 Problem

This project will solve a problem dealing with the current calibration process of this model data, as the current method does not allow for a visual comparison of the model output to the real-life data. In addition to solving this problem, the software will allow the user to edit the parameters of the model during this visualization. The improved calibration will help further the ecological understanding of climate change effects on plants and soil moisture. A challenge of creating the software will be fully understanding the entire workflow, as it requires understanding of complex ecological and meteorological concepts. NTSG, the sponsor’s organization, will benefit by streamlining the workflow and having an easily changeable code-base to improve model calibration for others to learn and continue to grow.

## 1.3 Desired Impact

With a cleaner calibration process of the climate change model, more accurate datasets will be created to make research with this data more reliable to accurately reflect real-life data. On a local scale, this new process will allow Arthur and other employees at NTSG to improve their workflow by being able to visually compare the model output to real life data. The impact that this improved calibration process will provide has the potential to be far-reaching on a global scale, such as the potential to allow scientists to better predict the effects of climate change on soil moisture and plant life. The understanding that this program will help provide will allow ecologists to be better prepared to face the consequences of climate change in the future.

# 2. Stakeholders

|  |  |
| --- | --- |
| ***STAKEHOLDER*** | ***Need/Problem Solved by Project*** |
| *Arthur* | *Arthur is the main individual, or primary stakeholder, that will be using this software. He currently uses the previously described calibration process at NTSG. The project will improve his job experience by being able to visually compare the model output to actual data and edit the model parameters.* |
| *NTSG Employees* | *NTSG employees, the secondary stakeholders or potential primary stakeholders, will indirectly and directly use this software. The program will improve user experience, which another employee may need to be trained on when learning the calibration process. Employees will also have to deal with the output of the model, which will be calibrated correctly with help of the program.* |
| *NTSG* | *NTSG will benefit from this program, because the new calibration process will improve the workflow by improving model data by packaging and improving the outputs. This company is a NASA-funded company so the data will be used by other NASA-funded companies and researchers.* |
| *Other Scientists* | *Other scientists will be dealing with the output of the model. This data can be used by more complicated models that scientists develop. This will further improve the understanding of climate change on soil moisture and plant life which will allow for the prediction of ecological effects.* |
| *Yolanda Reimer* | *Yolanda will be the grader of the project and not a user. She provides the structure of the entire program and overall guidance for our team. She will see an improvement in the knowledge of the team in both computer science and ecology areas.* |
| *Team members* | *We, as a team, will further the ability to problem solve by creating software in an area that we are not accustomed to working in. We will gain more understanding on how to communicate with clients and one another which we help us in our career. Experience in beginning to develop a program from the ground up will be useful in our careers as software developers.* |

# 3. Measurable Organizational Value

The Software shall improve the operational functions of NTSG by performing the climate simulations at least 20% faster and with 30% or more less memory than the previous method while having a modular design all by the end of May 2020.

# 4. Project Scope

All work within the scope of the project:

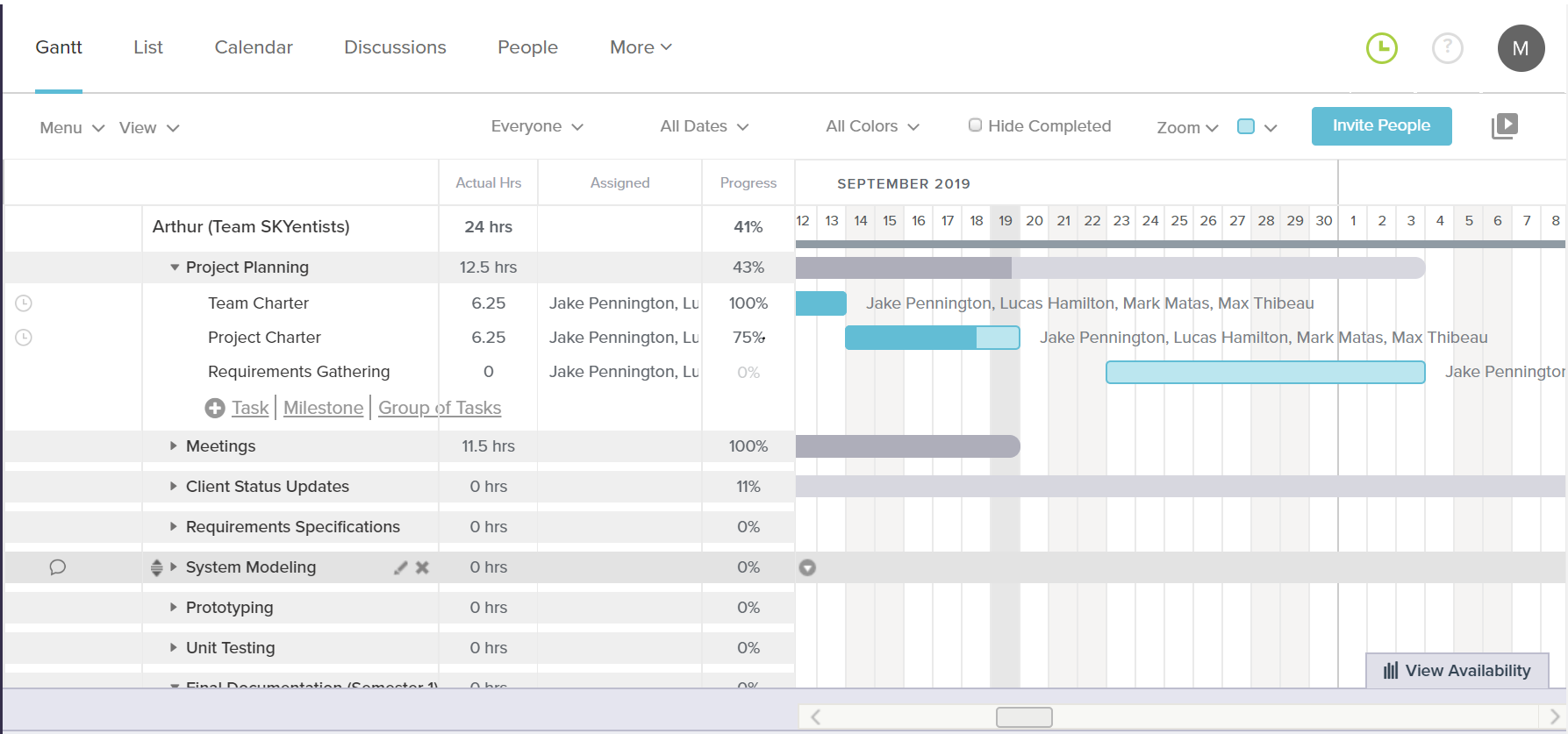
* Read in real-world data on plant growth and change
* Read in a model that simulates plant growth and change
* Show how the model currently predicts plant growth and change
* Update model behavior based on data
* Show the results of changing the model to better match the data
* Output a table of results that are used to run the model after calibration
* Calculate how well the model predicts part of the real-world data that was not used in calibration

All work outside the scope of the project:

* Create new data
* Create new models
* Create new calibration algorithms
* Edit existing MATLAB code to be more functional

# 5. Project Schedule

Include a printout of your TeamGantt chart showing major milestones for the semester.



# 6. Resources, Assumptions, & Risks

Most of the Resources that will be provided for this project will be provided by Arthur Endsley, which involve a finished version of a requirement/information documents that we will receive soon (we already received a rough draft of it on 9/16/19), matlab code that will be used to compare our future program against (received on 9/18/19), and inputs and data sets for calibration that are already provided on the NTSG website. Other resources (computers with certain specifications, workspace, basic tools) will either be provided personally or by the University of Montana within use by students of the University. More information on additional resources we may need will be more apparent later during the design process.

The Assumptions of the project will include:

That Arthur Endsley will be available for contact so the team may be informed of certain information when requested, discuss concerns in certain aspects of the project that may surface, or for requesting specific resources that cannot be obtained personally or by the University.\

That each team member will be available to work on their portion of the project each week and show up to regularly scheduled meetings, unless in emergency situations or situations discussed before the meetings are taking place.

The risks of the project will include:

The team not finishing the project in time, which would cause either Arthur Endsley to have to continue using his version of the calibration software and the company continuing to go without a clean modular system of software to use for their other projects, or for Arthur to find another team to finish the project in our place, which could result in possible loss of credit on project. Could also be damaging to the reputation of the SKYientists.

Our implementation not being faster and cleaner than what has already been provided, which could result in the same as not finishing the project in time.

Hardware failure which could result in loss of work done on the machine it is happening too and may result in the team member not being able to work on the project as efficiently.

The possible creation of Inaccurate data that could result in the hinderance of scientific progress.

The constrains of the project:

Hardware constraints: The project must be able to run on a Linux or MAC operating system. Should be able to run on Arthur’s computer.

Software constraints: Must use Python 3.xx with only libraries that are non-left and open source, with version control using GIT. Cannot emulate the matlab code we have been provided by Arthur.

Resource constraints: We can only use data that is provided to use by the NTSG website or by Arthur Endsley for the use in our software.

Outstanding issues:

There are currently no outstanding issues as of 09/19/18.

# 7. Acceptance & Approval

Team names, signatures, and dates of approval, including the client.

The Skyentists

 Mark Matas



Lucas Hamilton



Jake Pennington

Max Thibeau



Arthur Endsley (Client)

Final Project Charter

**Project Name:**

Earth System Models from Flux Tower Data

Prepared by:

The Skyentists

# Date:

9/19/19

# 1. Project Description and Problem

## 1.1 Background

This project involves creating a new calibration process for a climate change model that deals with different plant types and soil moisture. The model gets its test data from eddy flux towers located across the globe, as well as other scientific data. This improved calibration process will allow the user to tweak the model’s inputs and graph outputs to more accurately reflect the tower data. The sponsor of this project is the Numerical Terradynamic Simulation Group (NTSG), a group that researches tech development in certain Earth satellite data. The sponsor’s mission statement is "our primary mission is to develop capabilities to quantitatively describe the structure and function of ecosystems from regional to global scales using emerging technologies in satellite remote sensing, computational modeling and biophysical theory."

## 1.2 Problem

This project will solve a problem dealing with the current calibration process of this model data, as the current method does not allow for a visual comparison of the model output to the real-life data. In addition to solving this problem, the software will allow the user to edit the parameters of the model during this visualization. The improved calibration will help further the ecological understanding of climate change effects on plants and soil moisture. A challenge of creating the software will be fully understanding the entire workflow, as it requires understanding of complex ecological and meteorological concepts. NTSG, the sponsor’s organization, will benefit by streamlining the workflow and having an easily changeable code-base to improve model calibration for others to learn and continue to grow.

## 1.3 Desired Impact

With a cleaner calibration process of the climate change model, more accurate datasets will be created to make research with this data more reliable to accurately reflect real-life data. On a local scale, this new process will allow Arthur and other employees at NTSG to improve their workflow by being able to visually compare the model output to real life data. The impact that this improved calibration process will provide has the potential to be far-reaching on a global scale, such as the potential to allow scientists to better predict the effects of climate change on soil moisture and plant life. The understanding that this program will help provide will allow ecologists to be better prepared to face the consequences of climate change in the future.

**2. Stakeholders**

|  |  |
| --- | --- |
| ***STAKEHOLDER*** | ***Need/Problem Solved by Project*** |
| *The Client* | *The client is the main individual, or primary stakeholder, that will be using this software. Currently, the client uses the previously described calibration process at NTSG. The project will improve the job experience by being able to visually compare the model output to actual data and edit the model parameters.* |
| *NTSG Employees* | *NTSG employees, the secondary stakeholders or potential primary stakeholders, will indirectly and directly use this software. The program will improve user experience, by providing a GUI to perform the calibration process. Employees will also have to view and compare the output of the model to the real-life data, which will be calibrated correctly with help of the program.* |
| *NTSG* | *NTSG will benefit from this program, because the new calibration process will improve the workflow by improving model data by packaging and improving the outputs. This company is a NASA-funded company so the data will be used by other NASA-funded companies and researchers.* |
| *Other Scientists* | *Other scientists will be dealing with the output of the model. This data can be used by more complicated models that scientists develop. This will further improve the understanding of climate change on soil moisture and plant life which will allow for the prediction of ecological effects.* |
| *Professor Reimer* | *Professor Reimer will be the grader of the project and not a user. She provides the structure of the entire program and overall guidance for our team. She will see an improvement in the knowledge of the team in both computer science and ecology areas.* |
| *Team members* | *We, as a team, will further the ability to problem solve by creating software in an area that we are not accustomed to working in. We will gain more understanding on how to communicate with clients and one another which we help us in our career. Experience in beginning to develop a program from the ground up will be useful in our careers as software developers.* |

**3. Measurable Organizational Value(MOV)**

The Software shall improve the operational functions of NTSG by performing the climate simulations at least 20% faster and with 30% or more less memory than the previous method while having a modular design all by the end of May 2020.

**4. Project Scope**

All work within the scope of the project:

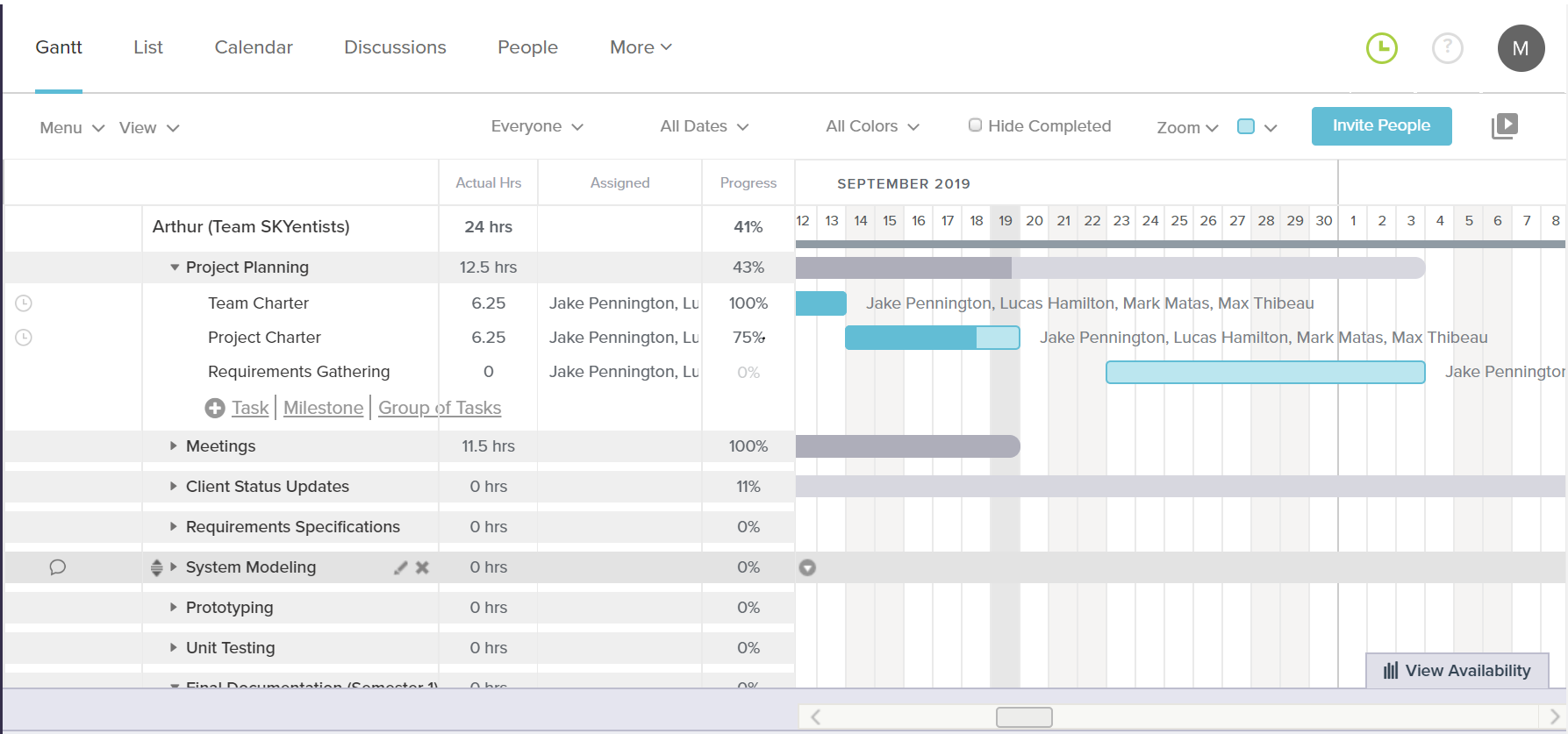
* Read in real-world data on plant growth and change
* Read in a model that simulates plant growth and change
* Show how the model currently predicts plant growth and change
* Update model behavior based on data
* Show the results of changing the model to better match the data
* Output a table of results that are used to run the model after calibration
* Calculate how well the model predicts part of the real-world data that was not used in calibration

All work outside the scope of the project:

* Create new data
* Create new models
* Create new calibration algorithms
* Edit existing MATLAB code to be more functional

**5. Project Schedule**

Include a printout of your TeamGantt chart showing major milestones for the semester.



**6. Resources, Assumptions, and Risks**

Most of the Resources that will be provided for this project will be provided by Arthur Endsley, which involve a finished version of a requirement/information documents that we will receive soon (we already received a rough draft of it on 9/16/19), matlab code that will be used to compare our future program against (received on 9/18/19), and inputs and data sets for calibration that are already provided on the NTSG website. Other resources (computers with certain specifications, workspace, basic tools) will either be provided personally or by the University of Montana within use by students of the University. More information on additional resources we may need will be more apparent later during the design process.

The Assumptions of the project will include:

That Arthur Endsley will be available for contact so the team may be informed of certain information when requested, discuss concerns in certain aspects of the project that may surface, or for requesting specific resources that cannot be obtained personally or by the University.\

That each team member will be available to work on their portion of the project each week and show up to regularly scheduled meetings, unless in emergency situations or situations discussed before the meetings are taking place.

The risks of the project will include:

The team not finishing the project in time, which would cause either Arthur Endsley to have to continue using his version of the calibration software and the company continuing to go without a clean modular system of software to use for their other projects, or for Arthur to find another team to finish the project in our place, which could result in possible loss of credit on project. Could also be damaging to the reputation of the SKYientists.

Our implementation not being faster and cleaner than what has already been provided, which could result in the same as not finishing the project in time.

Hardware failure which could result in loss of work done on the machine it is happening too and may result in the team member not being able to work on the project as efficiently.

The possible creation of Inaccurate data that could result in the hinderance of scientific progress.

The constrains of the project:

Hardware constraints: The project must be able to run on a Linux or MAC operating system. Should be able to run on Arthur’s computer.

Software constraints: Must use Python 3.xx with only libraries that are non-left and open source, with version control using GIT. Cannot emulate the matlab code we have been provided by Arthur.

Resource constraints: We can only use data that is provided to use by the NTSG website or by Arthur Endsley for the use in our software.

Outstanding issues:

There are currently no outstanding issues as of 09/19/18.

**7. Acceptance and Approval**

Team names, signatures, and dates of approval, including the client.

The Skyentists

 Mark Matas



Lucas Hamilton



Jake Pennington

Max Thibeau



Arthur Endsley (Client)

Old Requirements Specification Document

Prepared by: The Skyentists  
Due Date: 10/3/19

# 1. User Overview

The Client:

NASA expects the client to provide the Soil Moisture Active/ Passive (SMAP) Level 4 Carbon(L4C) data product. At a high level, the client would need to read in real world plant and soil data, calibrate the modeling software to match this data, and then simulate the SMAP L4C using these optimized parameters.

Other NTSG employees:

If the client were unable to produce the SMAP L4C, then other employees at NTSG can produce the data product. They would have to learn how the calibration process works and go through all the tasks that the client would perform.

Future Climatologists:

If NTSG were to shutdown and climatologists still wanted the SMAP L4C, they would have to perform all the tasks other NTSG employees would perform but with less technical expertise.

# 2. Hierarchy [or Organization Chart] of Users

As demonstrated in Figure 1, climatologists may use the SMAP L4C to further improve other more complex climate change models. The SMAP L4C is provided by NASA, who contracts the Numerical Terradynamic Simulation Group (NTSG) to produce this dataset. Currently the client, an employee for this company, is in charge of calibrating NTSG’s climate change model If the client is unable to perform this calibration then other NTSG employees will do so.

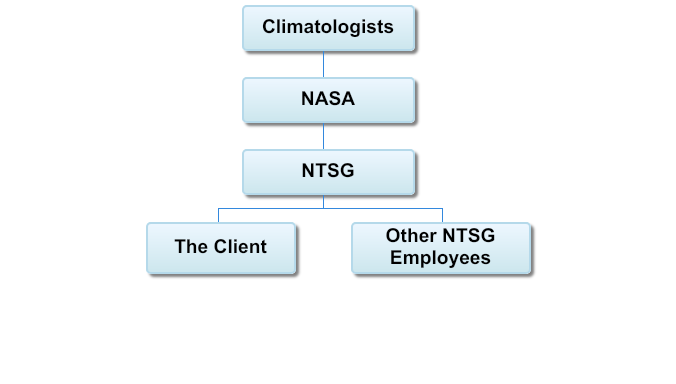


Figure 1. Hierarchy Chart

# 3. User Groups

As represented by Table 1, different user groups will use this software in different ways. The first column represents the name of the user group, the second column represents the number of users in that group, and the third column describes how the user group will be using this software.

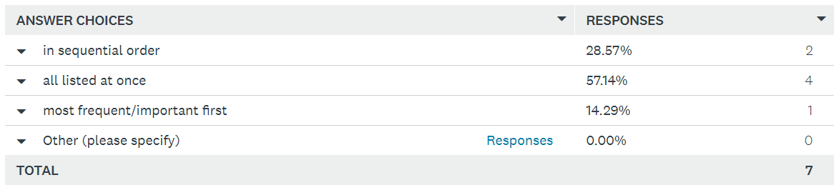
|  |  |  |
| --- | --- | --- |
| **Name** | **Number** | **Description** |
| The Client | 1 | The client will be the owner of the Github repository. He will use all features available from this software to collect data and calibrate models. This software will be used by the client daily. The Client has expert knowledge in this field. The Client will use a Linux machine. Other UN\*X machines could be used as well. The Client speaks English. The client will directly work with this software. |
| NTSG Employees | ~20 | NTSG Employees can clone and make pull requests from the Github repository but will not be able to directly edit the repository. They will be using the software for daily data collection and bi-annual model calibration if the client is unable to. NTSG employees have a moderate to high knowledge computer systems and the climatology of this project. The employees will be using UN\*X machines. They are all English speakers. They will work indirectly with the system when using the finished dataset, and directly with the system when generating it. |
| Climatologists | 1 Million+ | Other climatologists can clone and make pull requests for the repository, but will not be able to directly edit the repository. They will use all features of this software if NTSG is unable to. They will need to collect data daily and calibrate the model bi-annually. Computer system and application knowledge will vary between low and high expertise. Climatologists will use the web GUI to generate the dataset. Climatologists could speak any language. They will work indirectly with the system when using the finished dataset, and directly with the system when generating it. |

Table 1. User groups

# 4. User Feedback

We gathered user feedback by using a 4 question survey that was given to 7 of our DBS and CS (with modeling experience) peers. Here are the results of the survey:

Question 1: How would you prefer to select a certain category to edit?

Result:

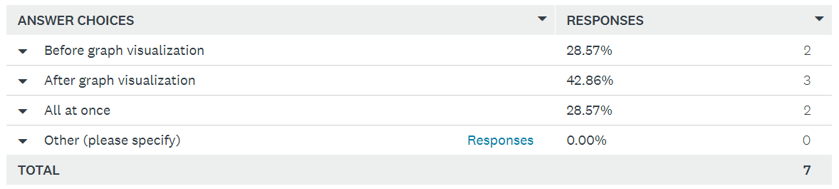
Summary: The majority of responders said they would prefer all choices be listed at once when selecting a certain category with a 57.14% vote.

Question 2: For editing the parameters for a model, would you rather edit each….

Result: 

Summary: The majority of the responders said they would prefer to edit parameters one at a time.

Question 3: When would you prefer to edit parameters?

Result: 

Summary: There was no clear majority with this poll. Most people preferred to edit parameters after graph visualization, but not by a large margin given the size of the sample. It seems that there is a little more subjectivity on when people want to edit parameters.

Question 4: Would you like to see what parameters are being used to calculate?

Result: 

Summary: All responders want to see what parameters are used for in the software.

# 5. Functional Requirements

Functional requirements define how the software will behave. Some functional requirements have prerequisites. Those prerequisites will be referenced at the end of those requirements in parentheses.

High Priority:

The following list displays the high priority functional requirements, or the most critical behaviors of the software. These range from directly affecting important business processes to improving workflow. These requirements will have, at most, one other high priority dependency in the software and will have lower priority dependencies.

The software shall:

* Calculate dominant Plant Functional Type (PFT) for each tower site
* Read in 4 input datasets (L4C reference dataset, L4C meteorological input dataset, Flux Tower Fluxes dataset, IGBP FAO Soil Organic Carbon Inventory dataset) for calibration
* Compile the L4C meteorological inputs and flux tower data into table
* Allow the user to choose one of the nine PFTs (High-1)
* Guide the user through removing outliers in average annual GPP and RECO calculations
* Allow user to choose which parameters to use when optimizing GPP and RECO
* Calculate linear ramp functions given current BLPUT (High-6)
* Allow user to specify number of Numerical Spin-Up iterations
* Compute comprehensive validation and fit statistics: graph flux tower data against model-estimated data
* Output updated BLPUT, 4 SOC stock-size maps, and SMRZ minimum and maximum for 2000-2018
* Calculate flux tower weights as some tower sites are located in same 9km section

Medium Priority:

The subsequent list shows the medium priority functional requirements. These are the software behaviors that directly or indirectly affect business flows. These requirements will be dependent on the high priority requirements and may have some low priority dependencies.

The software shall:

1. Calculate percentage of area that is occupied for each PFT at each tower site
2. Compile historical data and error metrics for each day for each PFT into table
3. Subset time series variables to user configurable period (High-2)
4. Subset L4C meteorological and L4C reference input to sites that have dominant selected PFT (High-4)
5. Compile ancillary info on each site for selected PFT such as dominant PFT at 9km scale and SOC stock size for the site (High-4)
6. Compile all important data for calibration of selected PFT that includes FPAR, PAR, TSURF, TSOIL, SMSF, VPD, TMIN, SMRZ, minimum SMRZ, maximum SMRZ, and PAW (High-4)
7. Average data for PFT each day of the year (High-5)
8. Display current ramp functions to the user and allow them to save the plots as files (High-7)
9. Allow the user to plot GPP against Emult separately form the ramp functions (Medium-8)
10. Report the differences between new and old value parameters after optimization (High-7)
11. Allow the user to specify Pk and Prh for RECO (High-6)
12. Create 2 plots: Rh/Cbar against TSOIL with the Arrhenius curve on top, Rh/Cbar against SMSF with the SMSF ramp function on top (High-7)
13. Calculate Cbar for each tower site after optimization (High-7, Medium-12)
14. Calculate σ and Bsoc for each tower site to plot σ \* Bsoc against ground truth SOC sizes
15. Run the Analytical and Numerical Model Spin-Ups (High-8)
16. Run preliminary spin up arbitrary period over full operational record 2000-2019
17. Run L4C soil model forward runs
18. Prepare vectors of initial optimized parameters and GPP and RECO optimization objective functions to iteratively change optimization parameters
19. Calculate GPP\*, Kmult\*, NPP\* based of calculations (Medium-7)

Low Priority:

The following list is the low priority functional requirements, which will be completed depending on time and the completion of the above two lists.

The software shall:

* Remove negative values in annual GPP and RECO for each flux tower site (Medium-7)
* Calculate lower and upper limits for APAR (Medium-8)
* Allow the user to repeat optimizations after the initial optimization (High-6, Medium-10)
* Allow the user to GPP against Emult and Rh/Cbar against Kmult

# 6. Non-functional Requirements

Non-functional requirements describe how the software should behave and how well the operator is able to use the program.

High Priority:

The requirements listed here are necessary for a minimum viable product. The project is considered a failure with the absence of any high priority requirement.

The Software shall:

1. Have accuracy that meets or exceeds the current implementation.
2. Be able to process data the size being used in the current implementation.
3. Pass a holistic test suite.
4. Be subjectively rated better than the current implementation by the client.
5. Follow a modular, object-oriented design pattern.
6. Have basic documentation for all features.

Medium Priority:

These requirements, while not integral to the system, will improve various aspects of the software.

The Software shall:

1. Have inline documentation for every class, method, and variable.
2. Run 20% faster than the current implementation.
3. Use 30% less memory than the current implementation.
4. Make use of Python virtual environments.
5. Have a test suite for every component.
6. Have sufficient maintainability as determined by the client.

Low Priority:

Requirements in this list have little to no impact on how the system will behave but add value to the project.

The Software shall:

1. Have a test case written for every method.
2. Have a test suite with 95% code coverage or higher.
3. Be validated on systems other than those specified in the software requirements.
4. Be validated on browsers other than those specified in the software requirements.

# 7. Hardware Requirements

The Software shall be developed on x86\_64 architecture with SSE4 extensions or later.

# 8. Software Requirements

* The Software shall be written in Python 3.
* The Software shall run on Unix and GNU/Linux based systems.
* The GUI shall run in a web browser (Mozilla Firefox or Google Chrome).

# Execution and Acknowledgement

The team members hereby indicate by their signatures below that they have read and agree with the specifications of this document.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | Lucas Hamilton, 10/13/19 |
| [Insert name & date here] |  | [Insert name & date here] |  | [Insert name & date here] |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Max Thibeau, 10/13/19 |  | Jake Pennington, 10/13/19 |  |  |
| [Insert name & date here] |  | [Insert name & date here] |  |  |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| [Insert Client name & date here] |  | [Insert Client name & date here] |  | [Insert Client name & date here] |

Final Requirements Specification Document

Prepared by: The Skyentists  
Due Date: 10/03/19  
Updated: 11/07/2019

Changes for Version 2

* Gave context to the questions asked to the users in User Feedback section
* Provided more in-depth explanation for some of the functional requirements
* Added better phrasing and some examples to some of the non-functional requirements
* All team member signatures were added
* All ethics certificates for the team members were added
* Added medium priority requirements based on client feedback

# 1. User Overview

The Client:

NASA expects the client to provide the SMAP L4C data product (SMLAP L4C). At a high level, the client would need to read in real world plant and soil data, calibrate the modeling software to match the ground truth data, and then simulate the SMAP L4C based off these optimized parameters.

Other NTSG employees:

If the client didn’t have the want or need to produce the SMAP L4C, then other employees at NTSG can produce the data product. They would have to learn how the calibration process works and go through all the tasks that the client would perform.

Future Climatologists:

If NTSG were to shut down and climatologists still wanted the SMAP L4C, they would have to perform all the tasks other NTSG employees would perform but with less technical expertise.

# 2. Hierarchy [or Organization Chart] of Users

As illustrated in Figure 1, climatologists may use the output of the calibration process to further improve other more complex climate change models. Most of the data is provided by NASA, who funds the Numerical Terradynamic Simulation Group (NTSG). Currently the client, an employee for this company, oversees calibrating NTSG’s climate change model. If the client is unable to perform this calibration, then other NTSG employees will use this software.

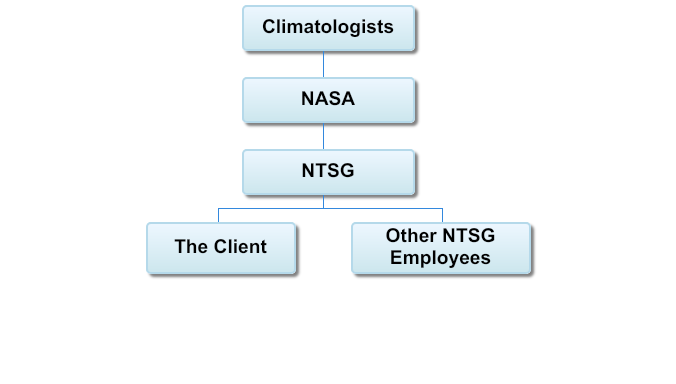


Figure 1. Hierarchy Chart

# 3. User Groups

Table 1.

This table represents the User Groups that will be using this software system. The first column represents the name of the user group, the second column represents the number of users that belong to said group, and the third column describes the details of how the User Group will be using the software system. The details revolve around what permissions they have, what they do at their current business, what features they will be using of the software system, how often they will be using the software system, what hardware/OS they will be using the software system on, their language, and if they will be interreacting with the system directly or indirectly.

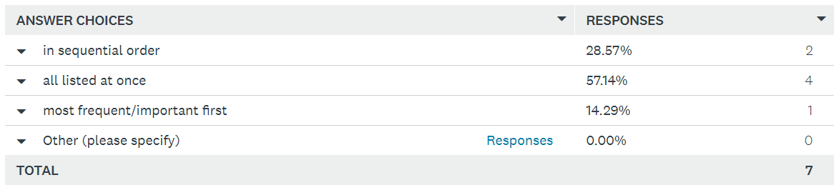
|  |  |  |
| --- | --- | --- |
| **Name** | **Number** | **Description** |
| The Client | 1 | The client will be the owner of the GIThub repository for the project, so he will control the repository settings and have administrative control over what gets pulled and pushed to the repository. The clients main work task is Earth System Model Calibration, so he will be using all of the features available from this software, which include the library of data structures and functions, The Flux Tower Observational Record Compiler, The PFT Selector and Calibrator, The Pre-Optimizer and Optimizer for GPP, The Pre-Optimizer and Optimizer for RECO, The SOC Parameter Fitter, The Analytical Model Spin-up, and The Numerical Model Spin-up (Approximation of features, subject to change later in development). This software will be used by the client daily. The Client appears to have a vast knowledge of Computer Systems and Application Domains. The Client will most likely being using a Linux machine, though could also be using other UN\*X machines as well. The Client speaks English. The Client will be interacting with the system directly. |
| NTSG Employees | Approximately 20 (Based on NTSG website) | The NTSG Employees will most likely be able to push and pull from the GIThub repository but will not be able to make changes to the repository settings. The NTSG employees work in the research of ecosystems using satellite information, so they will mostly be using the data structures and functions of the software rather than the software itself, which would probably vary between daily and weekly. However, they will be using the software bi-yearly for the calibration of the main model. The NTSG Employees will most likely have a moderate to high knowledge of computer systems and application domains. The employees will be using UN\*X machines due to the client specifying that the software is going to be built mainly for those machines in mind. Considering NTSG exists in UM, it is safe to say they are all English speakers. Interactions will vary dependent on what each employee is doing, so could be either direct or indirect. |
| Other Scientists | 1 Million+ | Other scientists will be able to pull from the repository, but will not be able to make changes, so if they want their own version, they will most likely be forking it too their own accounts. The tasks they preform during their business hours will vary. Considering how broad this group is, they could be using both the software and all its features along with the data structures and functions library of the software, but they will most likely on average be using the software only for special cases, so maybe every 5 years for the average scientist, but find more utility out of the data structures and functions library, which they will be using weekly at least if they have the need to use it. Computer system and application knowledge will most likely vary, averaging probably between moderate and high knowledge. Despite the system being made purely for UN\*X machines, people will probably find a way to use it on other OSs such as Windows. Since this system is made in America, it will more likely have more American users, so the majority of scientists will be English speakers, with a minority of scientists of other languages. Language shouldn’t be a problem, however, since this is an open source projects, so different versions in different languages can be created by others. They will most likely have a more indirect interaction with the system. |

# 4. User Feedback

The method we decided to use for user feedback was a 4 question survey that was given to 7 of our DBS and CS (with modeling experience) peers. Here are the results of the survey:

Question 1: How would you prefer to select a certain category to edit?

This question pertains to the calibration process, because the user will have to specify which plant function type (PFT) they are currently calibrating. There are nine total PFTs for this process so the user would choose the relevant PFT eight times.

Result:

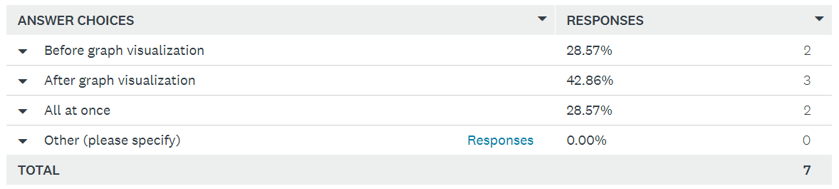
Summary: It seems that the majority of the responders said that they would prefer that all the choices be listed at once when selecting a certain category with a 57.14% vote. The follow up would be that people would want to select parameters in sequential order with a 28.57% vote. Only one person wants to select categories based on frequency and importance (14.29%). No one had any other options they wanted to specify.

Question 2: For editing the parameters for a model, would you rather edit each….

Result: 

Summary: The majority of the responders say they would prefer to edit parameters one by one as opposed to doing it all at once and grouping certain parameters together and then editing them. The second option people would settle for is “at once in one big grouping” with 28.57%. Only 1 person preferred editing in groups of certain parameters (14.29%).

Question 3: When would you prefer to edit parameters?

Result: 

Summary: There was a more even split on when the responders want to edit parameters. The highest percent goes to doing it after the graph visualization, though the two other options, before graph visualization and all at once, are close behind with the same percent. It seems that there is a little more subjectivity on when people want to edit parameters. However, no one had their own ideas on when to edit parameters.

Question 4: Would you like to see what parameters are being used to calculate?

The user would not have any control over the editing of the values for the relevant parameters, but only control which parameters are being optimized with the model data.

Result: 

Summary: All responders want to see the parameters that are being used to calculate with a perfect 100% on yes.

# 5. Functional Requirements

Functional requirements are the stipulations of how the software will behave. The functional requirements that the one of interest is dependent on is denoted inside the brackets () at the end.

High Priority:

The following list displays the high priority functional requirements, or the most critical behaviors of the software. These range from directly affecting important business processes to improving workflow. These requirements will have, at most, one other high priority dependency in the software and will have lower priority dependencies.

The software shall:

* Calculate dominant Plant Functional Type (PFT: Evergreen Needleleaf, Evergreen Broadleaf, Deciduous Needleleaf, Deciduous Broadleaf, Shrub, Grass, Cereal Crop, and Broadleaf Crop)for each tower site
* Read in 4 input datasets (L4C reference dataset, L4C meteorological input dataset, Flux Tower Fluxes dataset, IGBP FAO Soil Organic Carbon Inventory dataset) for calibration
* Compile the L4C meteorological inputs provided from NASA/NTSG and ground-truth data from flux tower sites into table
* Allow the user to choose one of the nine PFTs (High-1)
* Guide the user through removing outliers in average annual GPP and RECO calculations
* Allow user to choose which parameters to use when optimizing GPP and RECO, such as LUE (light use efficiency) for GPP or faut (autotrophic/plant respiration fraction) for RECO
* Calculate linear ramp functions given current BLPUT (High-6)
* Allow user to specify number of Numerical Spin-Up iterations
* Compute comprehensive validation and fit statistics: graph flux tower data against model-estimated data
* Output updated BLPUT, 4 SOC stock-size maps, and SMRZ minimum and maximum for 2000-2018
* Calculate flux tower weights as some tower sites are located in same 9km section

Medium Priority:

The subsequent list shows the medium priority functional requirements. These are the software behaviors that directly or indirectly affect business flows. These requirements will be dependent on the high priority requirements and may have some low priority dependencies. They are still vital in the calibration process itself, but are dependent on the high priority requirements

The software shall:

1. Calculate percentage of area that is occupied for each PFT at each tower site
2. Compile historical data and error metrics for each day for each PFT into table
3. Subset time series variables to user configurable period (High-2)
4. Subset L4C meteorological and L4C reference input to sites that have dominant selected PFT (High-4)
5. Compile ancillary info on each site for selected PFT such as dominant PFT at 9km scale and SOC stock size for the site (High-4)
6. Compile all important data for calibration of selected PFT that includes FPAR, PAR, TSURF, TSOIL, SMSF, VPD, TMIN, SMRZ, minimum SMRZ, maximum SMRZ, and PAW (High-4)
7. Average data for PFT each day of the year (High-5)
8. Display current ramp functions to the user and allow them to save the plots as files (High-7)
9. Allow the user to plot GPP against Emult separately form the ramp functions (Medium-8)
10. Report the differences between new and old value parameters after optimization (High-7)
11. Allow the user to specify Pk and Prh for RECO (High-6)
12. Create 2 plots: Rh/Cbar against TSOIL with the Arrhenius curve on top, Rh/Cbar against SMSF with the SMSF ramp function on top (High-7)
13. Calculate Cbar for each tower site after optimization (High-7, Medium-12)
14. Calculate σ and Bsoc for each tower site to plot σ \* Bsoc against ground truth SOC sizes
15. Run the Analytical and Numerical Model Spin-Ups (High-8)
16. Run preliminary spin up arbitrary period over full operational record 2000-2019
17. Run L4C soil model forward runs
18. Prepare vectors of initial optimized parameters and GPP and RECO optimization objective functions to iteratively change optimization parameters
19. Calculate GPP\*, Kmult\*, NPP\* based off of calculations (Medium-7)
20. Calculate RMSE for each subset of towers for observed both GPP and RECO against L4C Reference Dataset, optimized GPP/RECO, and L4C Forward Model Run (High-9)
21. Calculate NEE (NEE= RECO – GPP) based on optimized parameters then report stats of NEE against L4C Reference Dataset, optimized GPP/RECO, and L4C Forward Model Run

Low Priority:

The following list is the low priority functional requirements, which will be completed depending on time and the completion of the above two lists.

The software shall:

* Remove negative values in annual GPP and RECO for each flux tower site (Medium-7)
* Calculate lower and upper limits for APAR (Medium-8)
* Allow the user to repeat optimizations after the initial optimization (High-6, Medium-10)
* Allow the user to plot GPP against Emult and Rh/Cbar against Kmult

# 6. Non-functional Requirements

Non-functional requirements describe how the software should behave and how well the operator is able to use the program. All requirements listed below are numbered and shall be referenced by the first letter of their priority followed by their index (e.g. the first high priority requirement is H1).

High Priority:

The requirements listed here are necessary for a minimum viable product. The project is considered a failure with the absence of any high priority requirement.

The Software shall:

1. Have accuracy of calibration that meets or exceeds the current implementation.
2. Be able to process data while maintaining the size being used in the current implementation.
3. Pass a holistic test suite (a script that individually and automatically tests the functionality of the entire project, such as Python’s built-in “unittest”).
4. Be subjectively rated better than the current implementation by the client.
5. Follow a modular, object-oriented design pattern.
6. Have basic documentation for all features.

Medium Priority:

These requirements, while not integral to the system, will improve various aspects of the software.

The Software shall:

1. Have well-written inline documentation for every class, method, and variable (the code itself).
2. Allow the process, as a whole, to run 20% faster than the current implementation.
3. Use 30% less total, saved memory than the current implementation.
4. Make use of Python virtual environments.
5. Have a test suite for every component.
6. Have sufficient maintainability as determined by the client.

Low Priority:

Requirements in this list have little to no impact on how the system will behave but add value to the project.

The Software shall:

1. Have a test case written for every method.
2. Have a test suite with 95% code coverage or higher.
3. Be validated on systems other than those specified in the software requirements.
4. Be validated on browsers other than those specified in the software requirements.

# 7. Hardware Requirements

The Software shall be developed on x86\_64 architecture with SSE4 extensions or later.

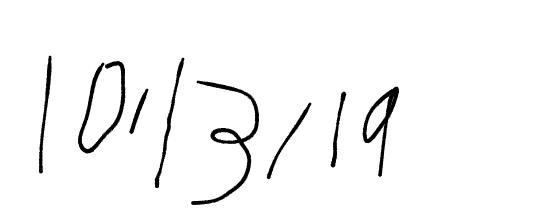
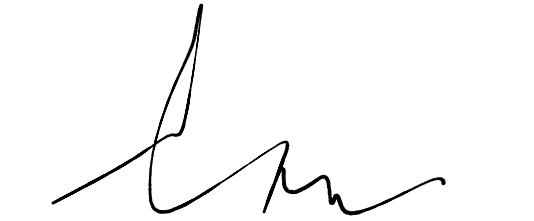
# 8. Software Requirements

* The Software shall be written in Python 3.
* The Software shall run on Unix and GNU/Linux based systems.
* The GUI shall run in a web browser (Mozilla Firefox or Google Chrome).

# 9. Execution and Acknowledgement

The team members hereby indicate by their signatures below that they have read and agree with the specifications of this document.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| [Insert name & date here] |  | [Insert name & date here] |  | [Insert name & date here] |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| [Insert name & date here] |  | [Insert name & date here] |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| [Insert Client name & date here] |  | [Insert Client name & date here] |  | [Insert Client name & date here] |

Old Systems Modeling Document

Prepared by: The Skyentists

Date: 10/17/19

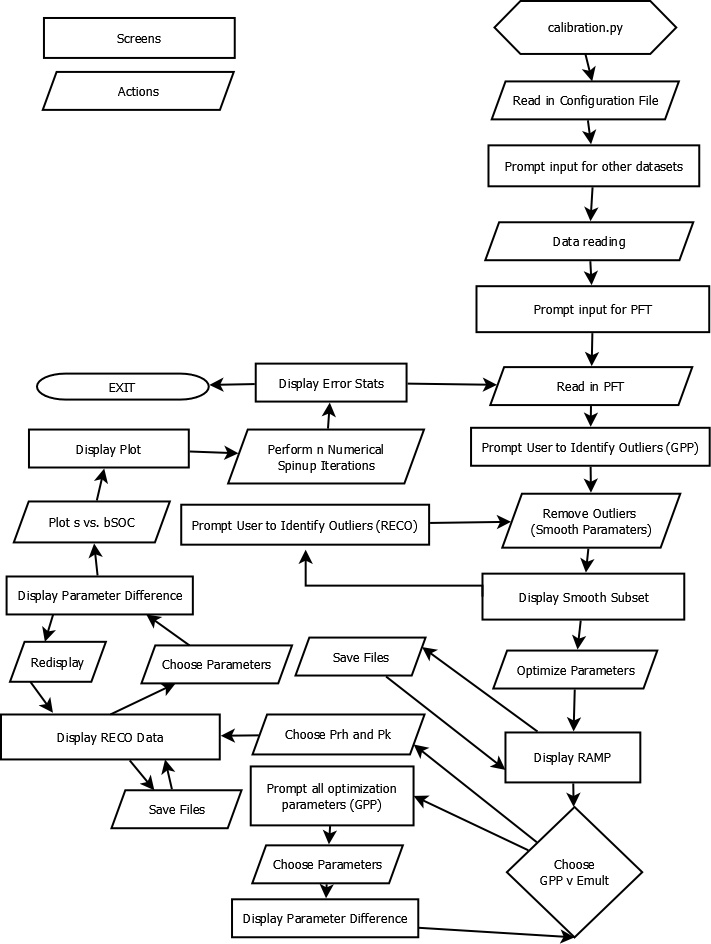
**1. Document Overview**

We create models in software development to partition complex processes into simple components. Attempting to program complex software without first modeling its behavior often leads to developers ‘coding themselves into a corner’. A systems model lets the user see the workflow of a system in an abstract manner, allowing for easy modifications without massive changes in codebase. Different models focus on different parts of a system. A commonly used model is Swimlane, a model that focuses on users interacting with other users in a system. Another popular model is the Entity Relationship model, which mimics an Object-Oriented programming approach with classes, attributes, and actions. The rest of the document contains two systems models, the first model focusing on a single user interacting with a system, and a second model focusing on data flow.

**2. Model 1 – State Transition Network**

The state transition network is a model that shows different states that the user will interact with during the calibration process. Each square box represents a display and each parallelogram represents an action the user can take. For the most part this diagram moves in one direction. The user inputs data necessary for calibration and the software uses that input accordingly. Occasionally there are small roundabouts in the workflow when the user has the choice to review changed plots or save figures. We chose this model to help visualize our user interface. Each square box represents a different screen we will need to design. Through making this diagram, we learned that most of the calibration software works in the backend, away from the user’s eyes. We also learned that each screen will have a simple display but, if designed poorly, will be hard to understand for those will little knowledge in climatology.

Diagram 1. This model displays the screens that the user will interact with throughout the software within a state transition network.



**3. Model 2 – Data Flow Diagram**

The Data Flow Diagram (DFD) is a type of model that demonstrates the ways the data flows throughout the system and the processes performed to alter this data. The model has a key of circles for processes, arrows for data flows, sandwiches for data stores, and rectangles for external entities. The DFD allows a high-level view on the data and allows for sub-models to be created for more complex functions. This model provides an accurate description of the calibration process due to the program having a plethora of functions and back-end work to transform the data. A DFD also goes with the functional requirement analysis since it is easy to spot missing functions or requirements that need to be performed on the data.

When creating the DFD, the team learned that the data will be going through many complex processes and data stores during the full calibration process. There was also the identification of functions more complex than previously thought throughout the model creation. After creation and discussion, there are less data stores than were present in the first draft of the model.

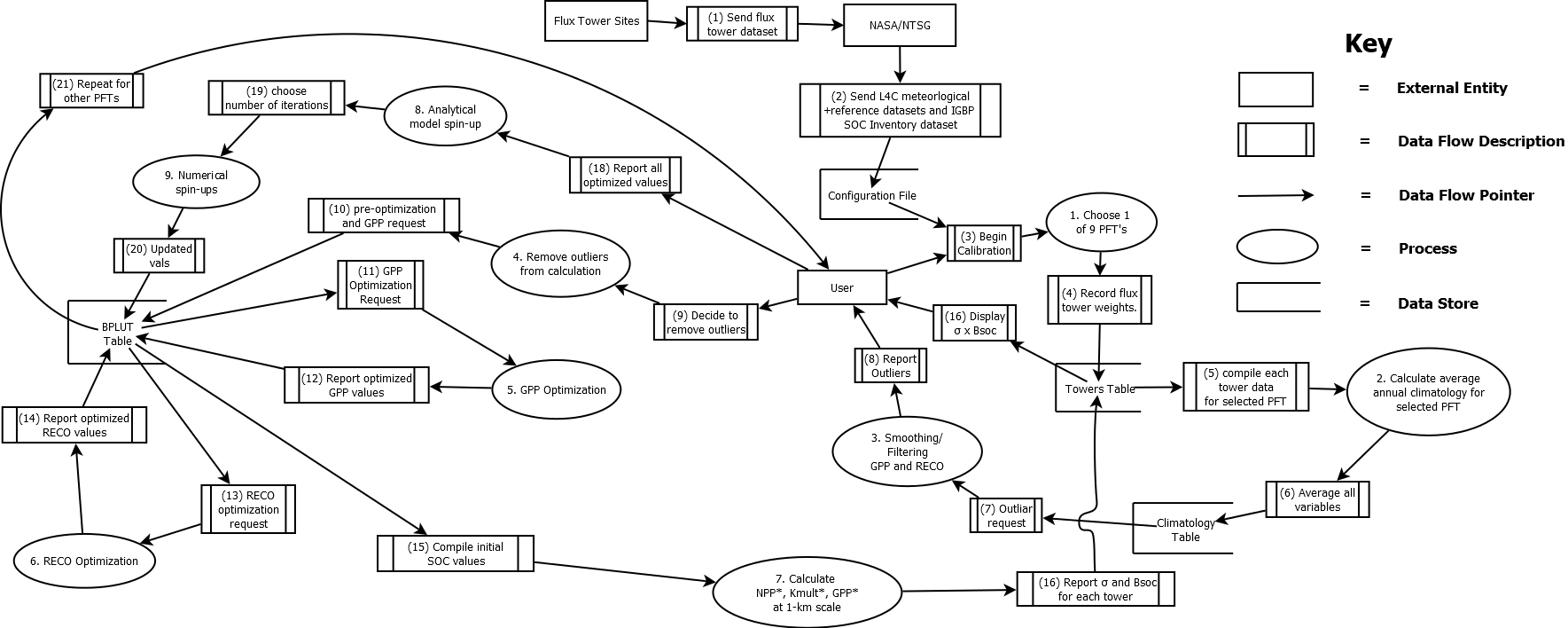
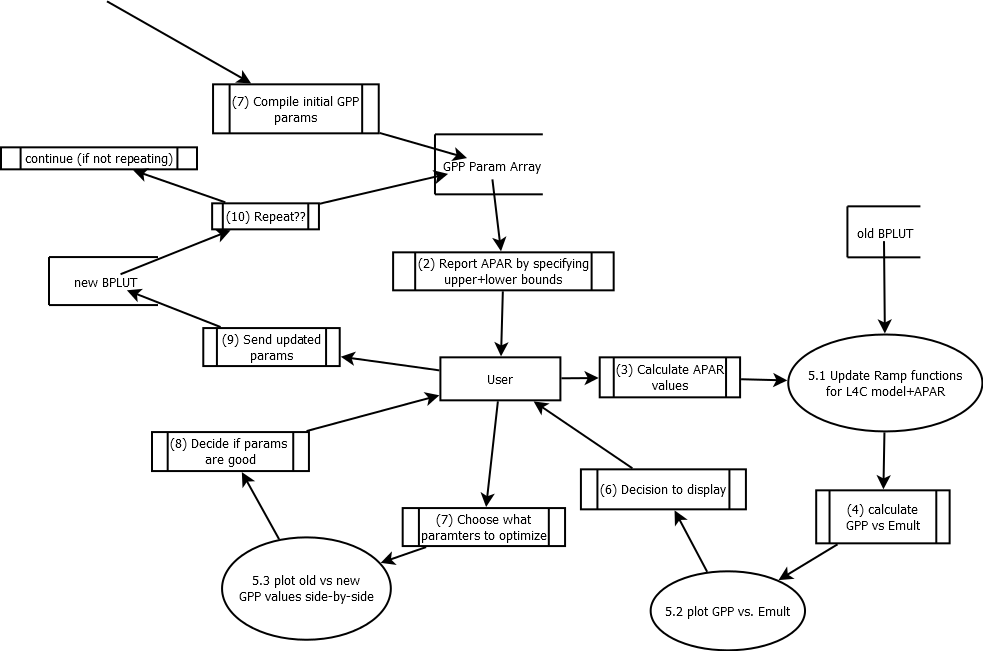
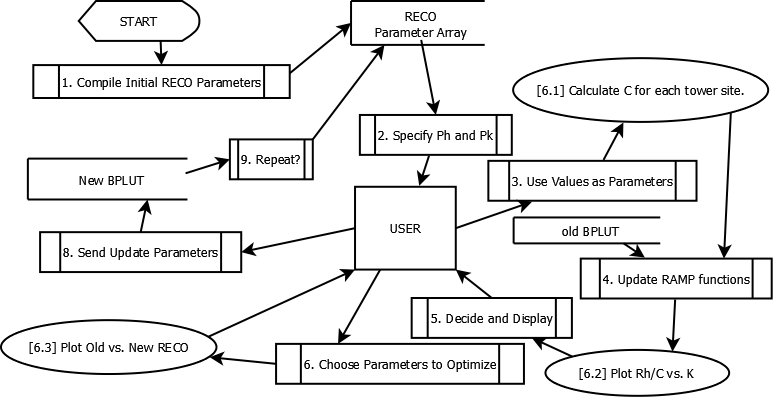
Diagram 2. This diagram shows the highest-level view of the calibration process in a data flow diagram. This details the processes that are performed on the data from the start of the process to the end of the process.

Diagram 3. This diagram displays the GPP optimization process from the beginning step, compiling the initial GPP parameters, to the final step, sending off the optimized GPP parameters to RECO optimization process (number 5 on Diagram 2).



Diagram 4. This diagram displays the GPP optimization process from the beginning step, compiling the initial GPP parameters, to the final step, sending off the optimized GPP parameters to RECO optimization process (number 5 on Diagram 2).

**4. Execution and Acknowledgement**

The team members hereby indicate by their signatures below that they have read and agree with the specifications of this document.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| [Insert name & date here] |  | [Insert name & date here] |  | [Insert name & date here] |





|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| [Insert name & date here] |  | [Insert name & date here] |  |  |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| [Insert name & date here] |  | [Insert Client name & date here] |  | [Insert Client name & date here] |

Final Systems Modeling Document

Prepared by: The Skyentists

Date: 10/17/19

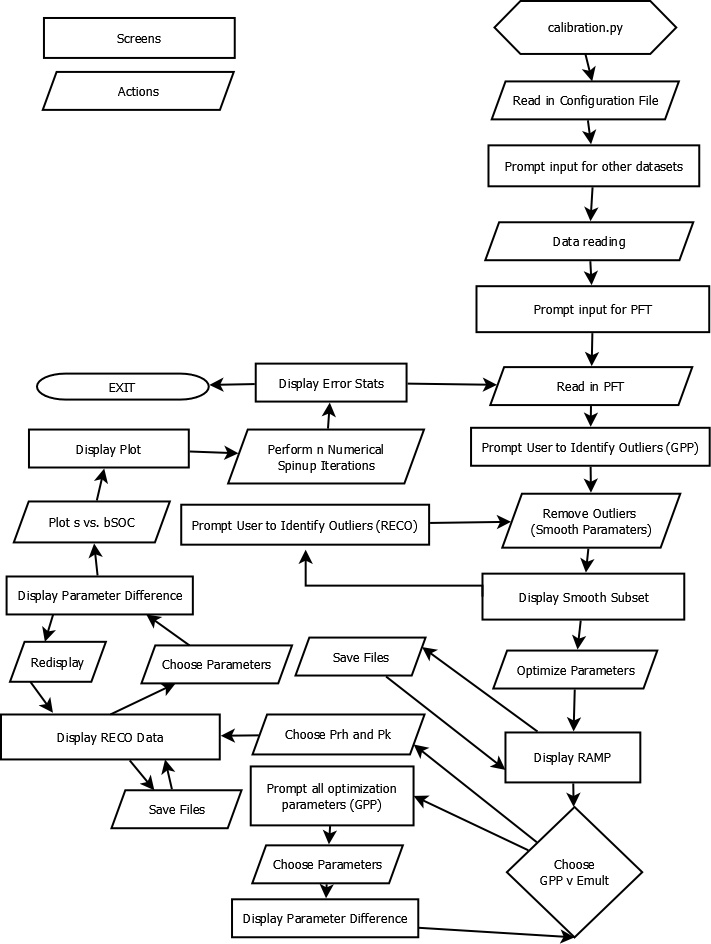
**1. Document Overview**

We create models in software development because workflows can be too complex to design through coding alone. If software developers jump right into coding as soon as they understand the requirements of a system, they could code themselves into a corner if they poorly designed their workflow. A systems model lets the user see the workflow of a system in an abstract manner, allowing for easy modifications without massive changes in codebase. Different models focus on different parts of a system as they complete a business task or process. A commonly used model is Swimlane which focuses on users interacting with other users in a system. Another popular model is the Entity Relationship model, which mimics an Object-Oriented programming approach with classes, attributes, and actions. The rest of the document contains two systems models with the first model, a State Transition Network, focusing on a single user interacting with a system and a second model, a Data Flow Diagram, that focuses on data flow.

**2. Model 1 – State Transition Network**

The state transition network shown in Diagram 1 is a model that shows the different states that the user will interact with during the calibration process. This calibration process helps the user edit the values of the parameters that will be used for in an advanced climate change model. Each square box represents a display and each parallelogram represents an action the user can take. For the most part this diagram moves in one direction. The user inputs data necessary for calibration and the software uses that input accordingly. Occasionally there are small roundabouts in the workflow when the user has the choice to review changed plots or save figures. We chose this model to help visualize our user interface. Each square box represents a different screen we will need to design. We will have to design many displays, but each display seems relatively simple to create. Through making this diagram, we learned that most of the calibration software works in the backend, away from the user’s eyes. We also learned that each screen will have a simple display, but if designed poorly will be hard to understand for those will little knowledge in climatology.

Diagram 1. This model displays the screens that the user will interact with throughout the software within a state transition network.



**3. Model 2 – Data Flow Diagram**

The Data Flow Diagram (DFD) is a type of model that demonstrates the ways the data flows throughout the system and the processes performed to alter this data. The model has a key of circles for processes, arrows for data flows, sandwiches for data stores, and rectangles for external entities. The DFD allows a high-level view on the data and allows for sub-models to be created for more complex functions. This model provides an accurate description of the calibration process due to the program having a plethora of functions and back-end work to transform the data. A DFD also goes hand-in-hand with the functional requirement analysis since it is easy to spot missing functions or requirements that need to be performed on the data.

When creating the DFD, the team learned that the data will be going through many complex processes and data stores during the full calibration process. There was also the identification of more complex functions than previously thought throughout the model creation. After creation and discussion, there are less data stores than were present in the first draft of the model.

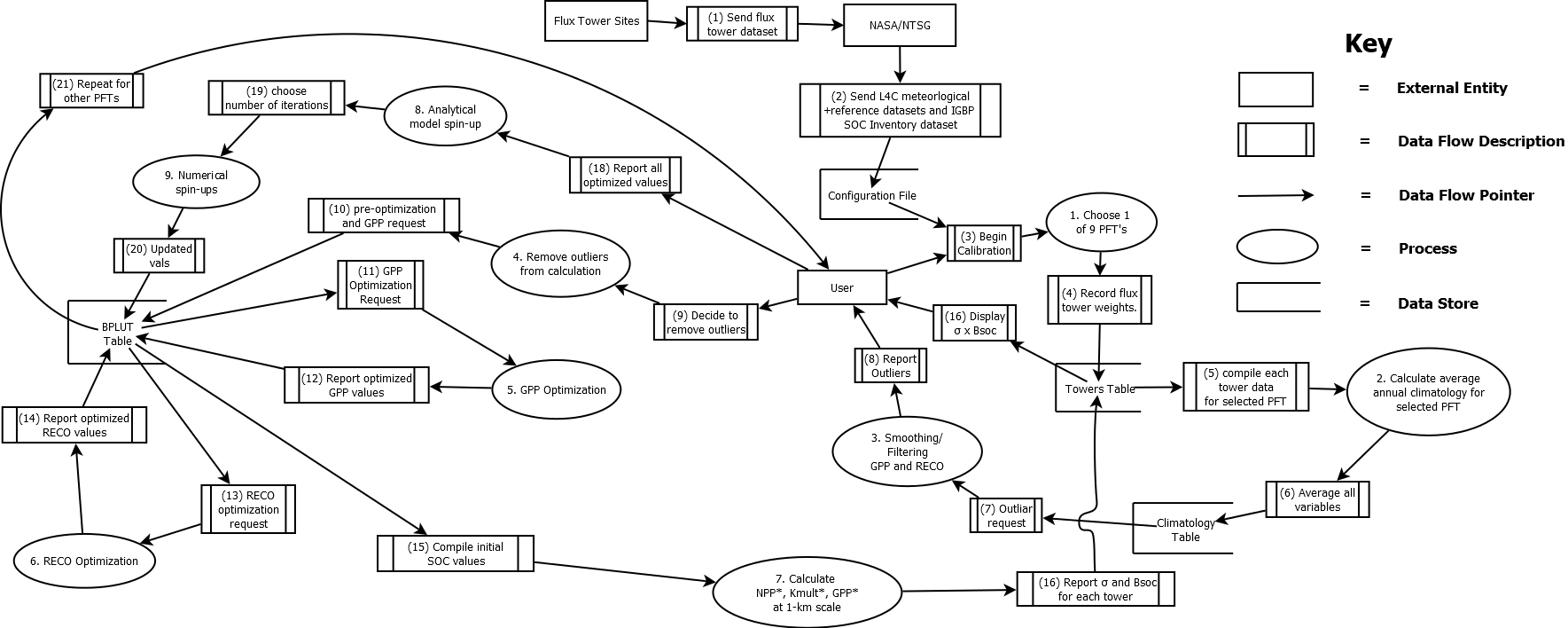
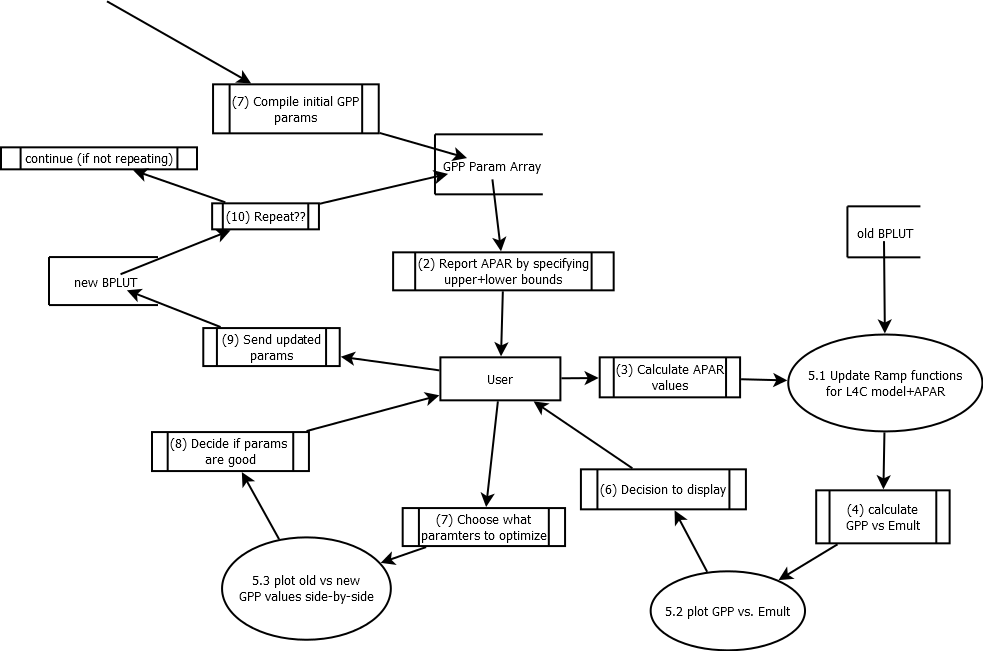
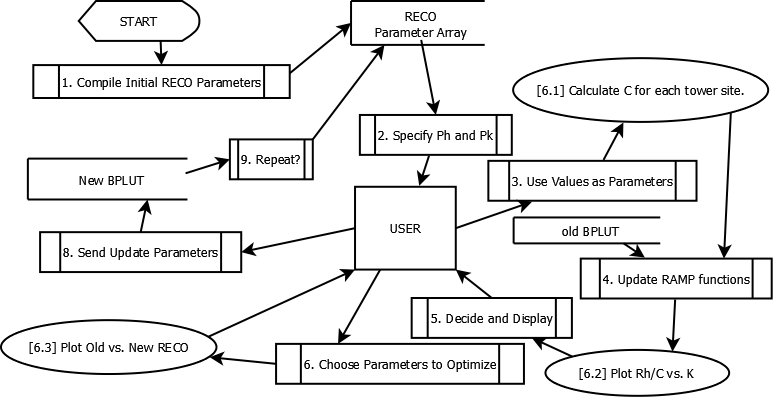
Diagram 2. This diagram shows the highest-level view of the calibration process in a data flow diagram. This details the processes that are performed on the data from the start of the process to the end of the process.

Diagram 3. This diagram displays the GPP optimization process as a whole from the beginning step, compiling the initial GPP parameters, to the final step, sending off the optimized GPP parameters to RECO optimization process (number 5 on Diagram 2).



Diagram 4. This diagram displays the RECO optimization process, similar to GPP optimization in Diagram 3. The entire RECO process from initializing parameters to sending off the optimized RECO parameters to the BPLUT table (number 5 on Diagram 2).

**4. Execution and Acknowledgement**

The team members hereby indicate by their signatures below that they have read and agree with the specifications of this document.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| [Insert name & date here] |  | [Insert name & date here] |  | [Insert name & date here] |





|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| [Insert name & date here] |  | [Insert name & date here] |  |  |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| [Insert Client name & date here] |  | [Insert Client name & date here] |  | [Insert Client name & date here] |

Old Low-fidelity

Prototypes

Document

Prepared by: The Skyentists  
Due Date: 10/24/19

**1. Overview**

UI prototypes take the user through the process of using software without fully implementing the software. UI prototypes consist of rough drafts of the software and have a limited degree of interaction. This is critical to developing an effective UI as it is the most cost and time efficient way to get feedback from users. Developing software is expensive, and it’s even more expensive to re-implement a finished product because the old one had a poor user interface.

‘

There are several different types of prototypes. The first type of prototype implemented in the software development process is a low fidelity prototype. Low fidelity prototypes are typically hand drawn sketches of the software process. These have the advantages of being low cost and easy to edit.

Low fidelity prototypes have a limited degree of interaction, which is why they serve as a benchmark for the other main UI prototype: the high fidelity prototype. High fidelity prototypes are realistic depictions of a program with a high degree of interactivity. High fidelity prototypes lets the user give feedback on all parts of the software, especially edge cases. Users can give more meaningful feedback as well, as they can interact with the prototype as if it were the software. All of these benefits do come at a higher cost than the low fidelity prototype.

Medium level prototypes are a blend between low and high fidelity prototypes. They are more interactive and expensive than low level prototypes, but not nearly as in depth and costly as high fidelity prototypes.

**2. Low Fidelity Prototypes**

The high priority functional requirements that are shown by the screenshots below are as followed:

1. Calculate dominant Plant Functional Type (PFT) for each tower site
2. Read in 4 input datasets (L4C (Level 4 Carbon) reference dataset, L4C meteorological input dataset, Flux Tower Fluxes dataset, IGBP FAO (Our empirical one) Soil Organic Carbon Inventory dataset) for calibration
3. Compile the L4C meteorological inputs and flux tower data into a table
4. Allow the user to choose one of the nine PFTs
5. Guide the user through removing outliers in average annual GPP (Gross Primary Production i.e. plants absorbing CO2) and RECO (Ecosystem Respiration Net CO2 output by plants and soil) calculations
6. Allow user to choose which parameters to use when optimizing GPP and RECO
7. Calculate linear ramp functions given current BPLUT (Biome Properties LookUp Table: All the data we have for a single plant type)
8. Allow user to specify number of Numerical Spin-Up iterations and then review results
9. Compute comprehensive validation and fit statistics: graph flux tower data against model-estimated data
10. Output updated BLPUT, 4 SOC (Soil Organic Carbon: CO2 emitted by soil) stock-size maps, and SMRZ (Soil Moisture Root Zone: amount of moisture plants roots can absorb) minimum and maximum for 2000-2018
11. Calculate flux tower weights (some tower sites are in the same 9km section)

A picture containing text

Description automatically generated

Intro Page

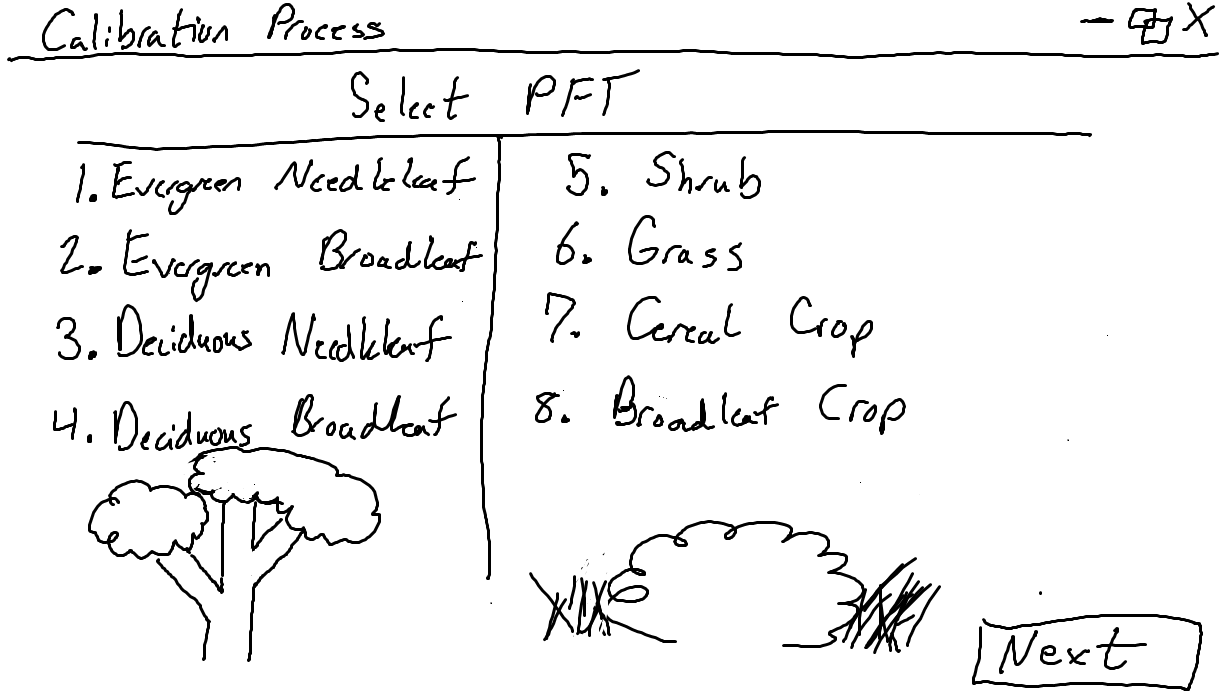
This page shows the first screen of the calibration process. Upon clicking the “Begin Calibration” button, the process of calibrating the SMAP L4C algorithm will start.

A close up of text on a white background

Description automatically generated

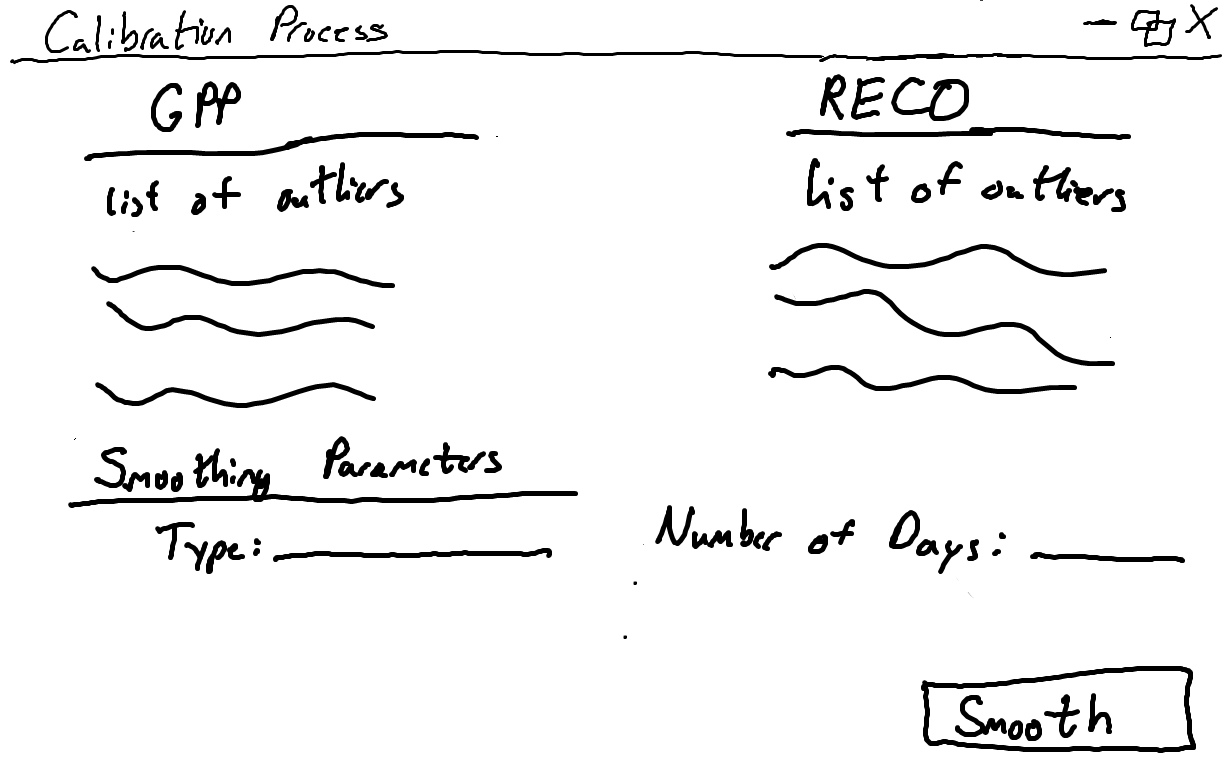
Configuration File Page

This page displays the name of the configuration file that the user will select. The configuration file consists of the current BPLUT table, list of flux towers sites to include and exclude, file path to the previously used Nature Run dataset, file paths for the HDF5 inputs, and the file path for the output file. The ‘Next’ button will allow the reading in of the four input datasets (High priority #2) and the compiling of the meteorological inputs and flux tower data into a single table (High priority #3).



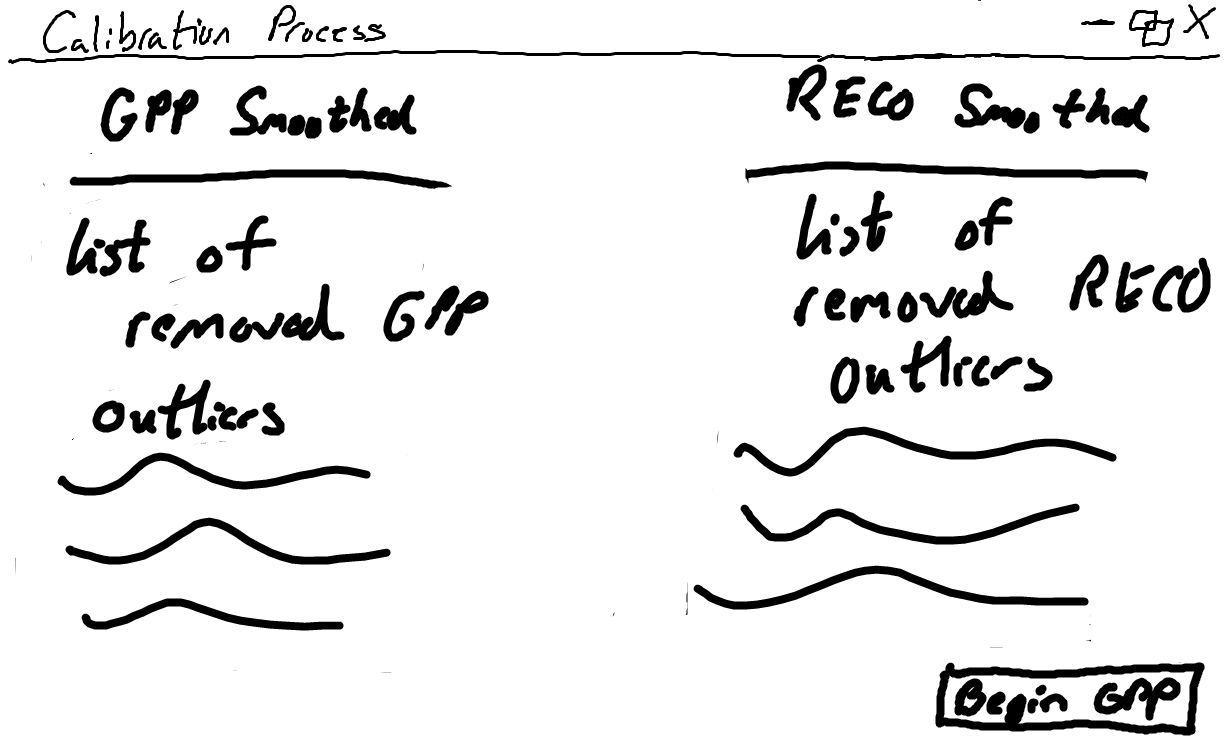
PFT Selection Page

This page lets the user choose 1 of 8 plant functional types (PFTs, high Priority #4). This PFT selection will determine the following ramp function calculations and optimization, as each one will have its own subset of tower data (High priority #1). The selected PFT will be highlighted and the ‘Next’ button will be enabled.



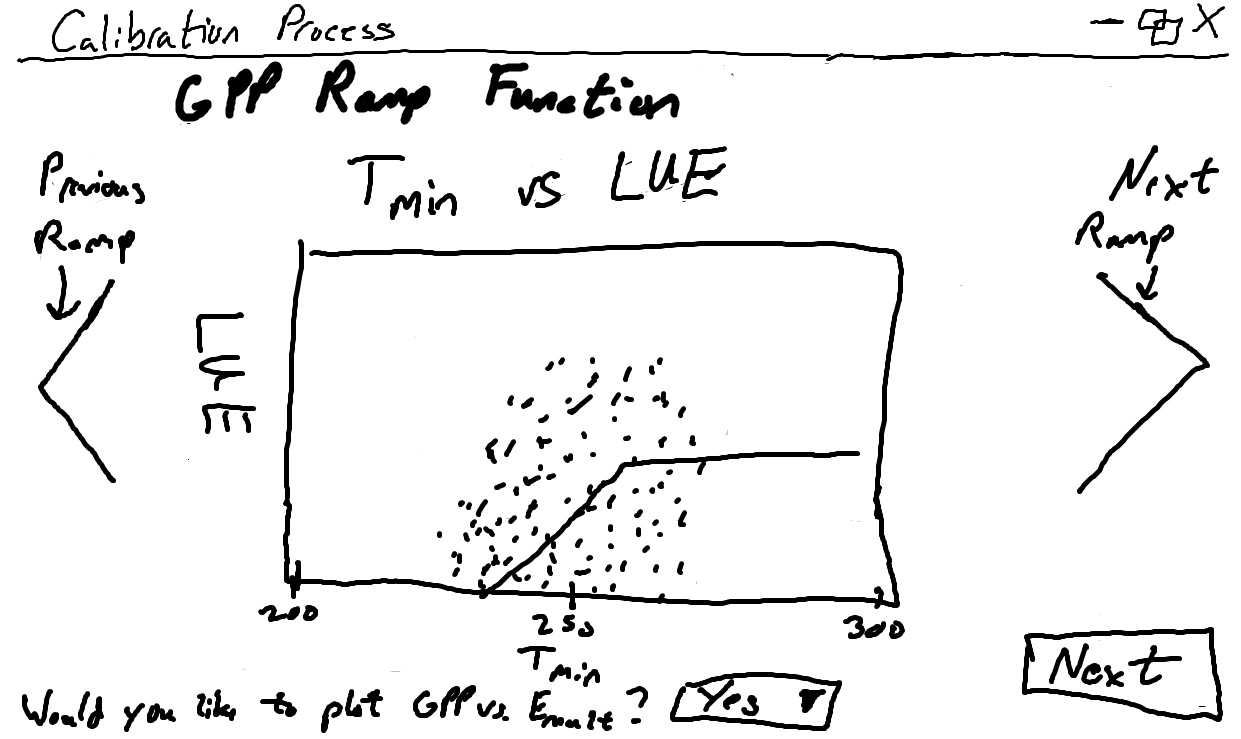
Selecting Outliers Page

This page allows the user to choose outliers in the average annual GPP and RECO that can be removed (High priority #5). The user will also be able to choose the type of smoother to use, such as SciPy’s filtfilt() implementation, and specify the day range of the dataset. Clicking the ‘Smooth’ button will begin the smoothing function.



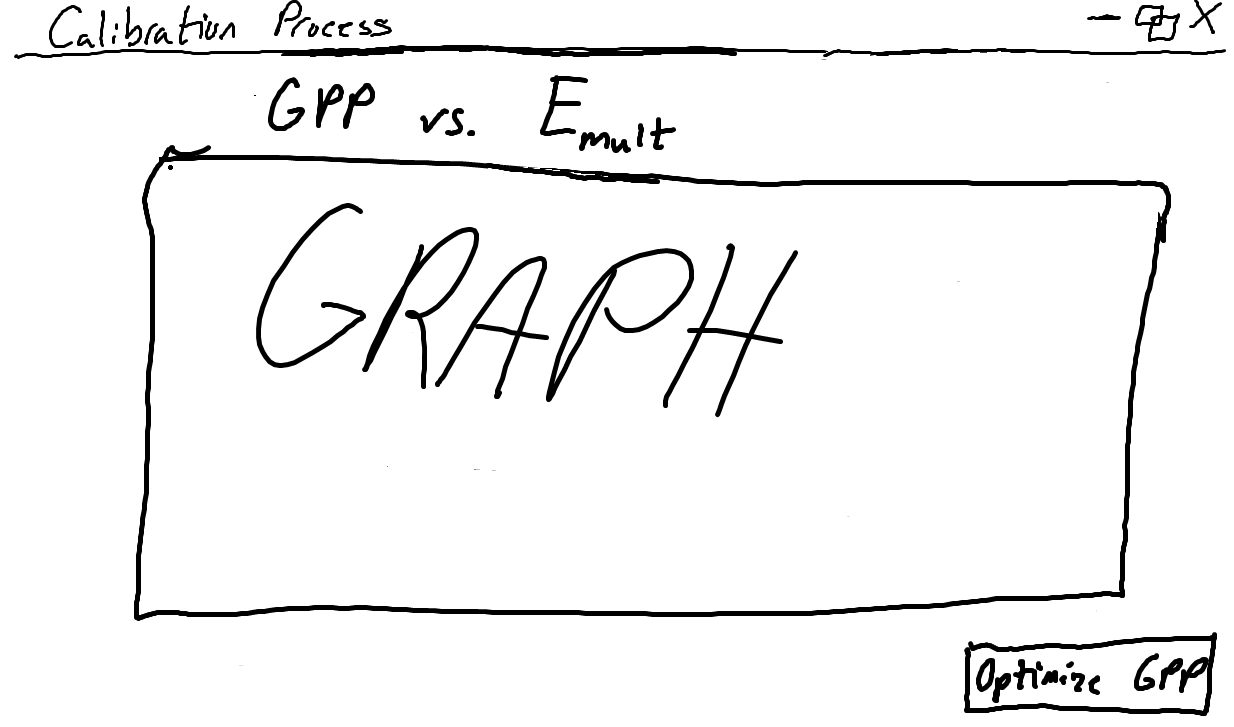
Removed Outliers Page

This page reports the selected outliers that are removed from the subset of data for the selected PFT due to the smoothing function. Clicking ‘Begin GPP’ will start the optimization function.



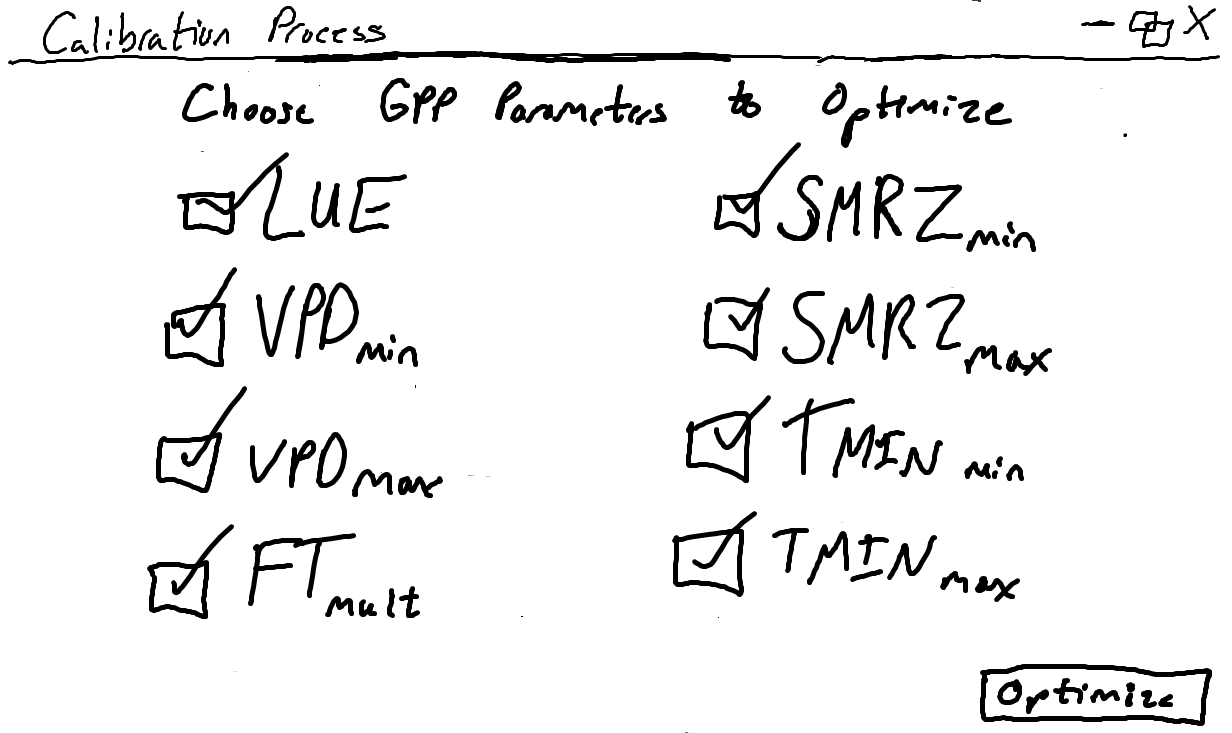
GPP Ramp Function Page

This page displays the plots of the current GPP ramp functions using the current BPLUT table (High priority #7), a part of the configuration file that includes data on the selected PFT. The option to display the GPP vs Emult graph is presented to the user with a dropdown for Yes or No, showing the below page with Yes selected when clicking the ‘Next’ button.



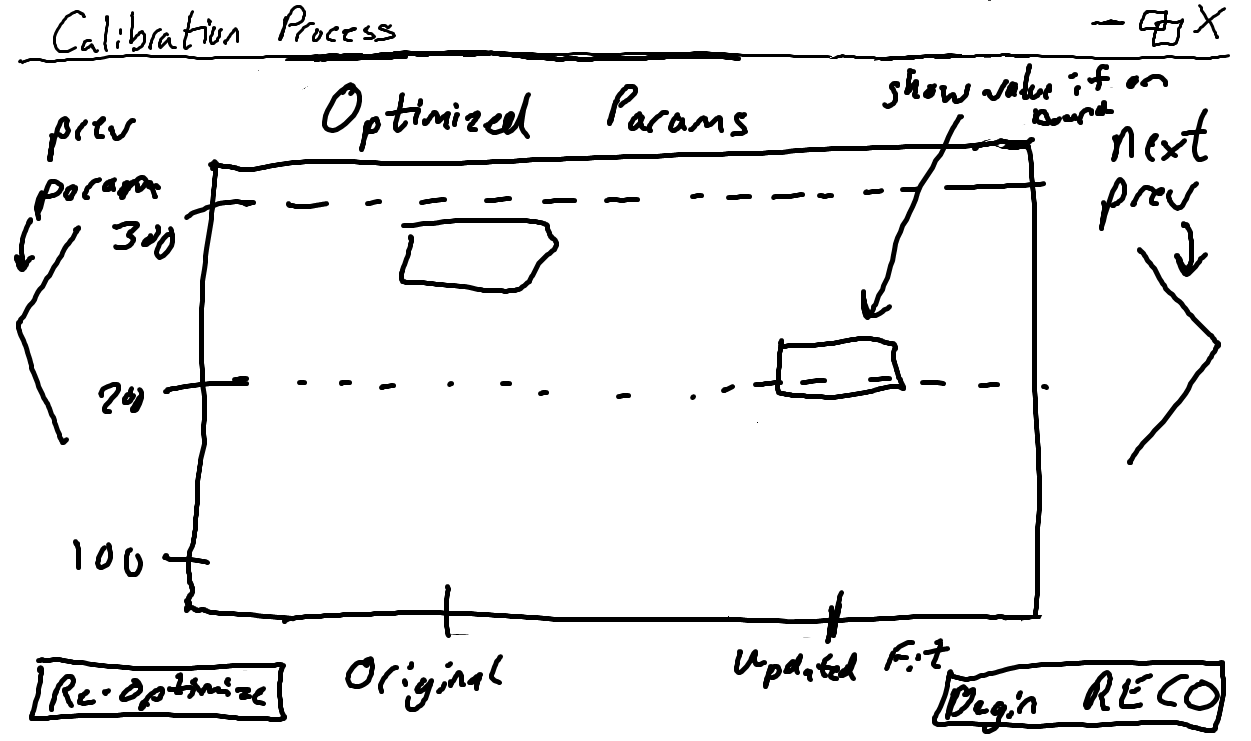
GPP vs Emult Graph Page

This page displays the graph of GPP vs. Emult which is an optional graph that the user may or may not wish to view. Clicking the ‘Optimize GPP’ button will start the optimization of GPP parameters.



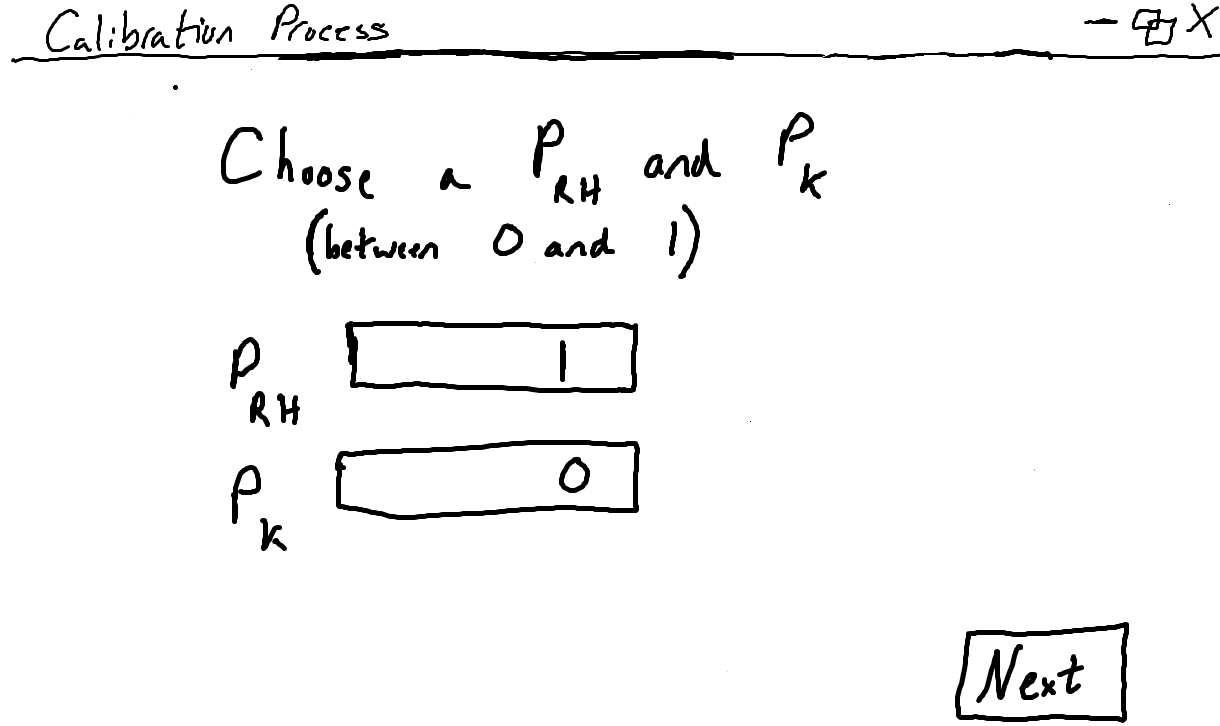
GPP Parameters Page

This page allows the user to choose which of the eight GPP parameters will be optimized (High priority #6). All eight of these parameters will automatically be checked as a default for the optimization process, but can be unchecked to be excluded. The ‘Optimize’ button begins the GPP optimization process.



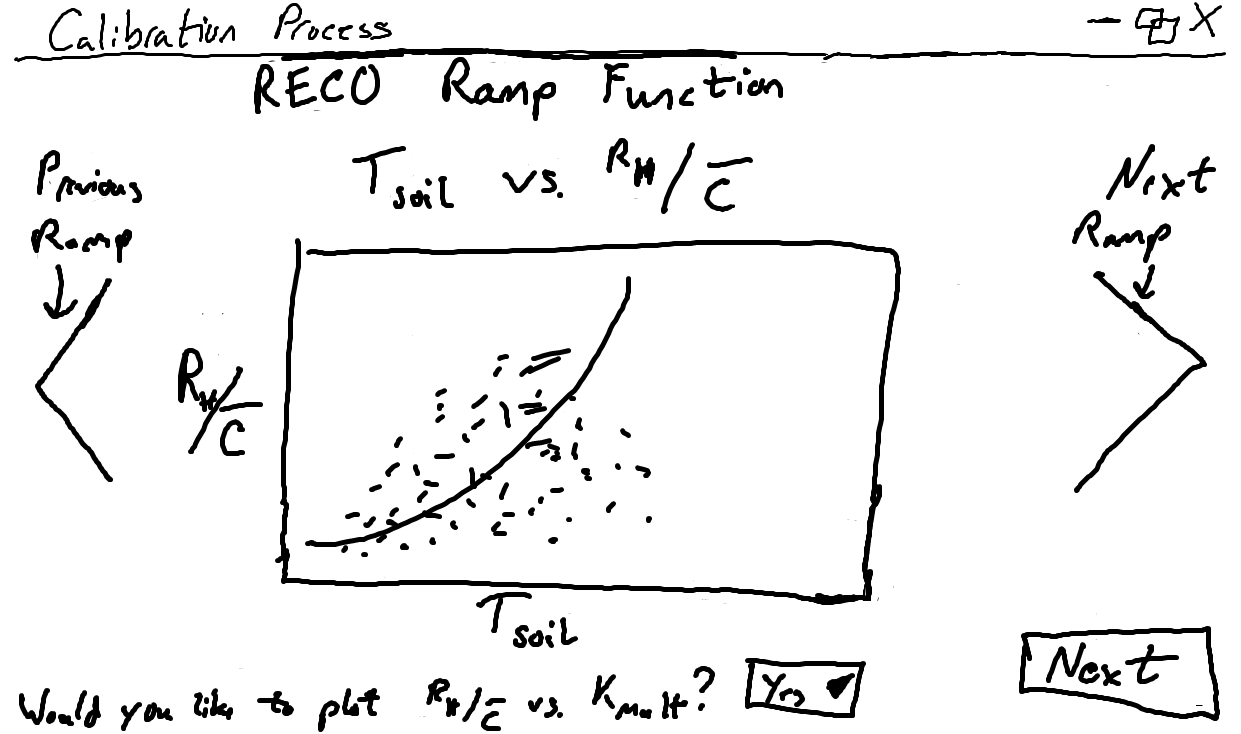
GPP Differences Page

This page displays both the original and updated fit for each optimized parameter. The value of the updated fit will be reported if it falls on or above/below the upper or lower bound. The updated fit is stored and updated in the BPLUT table. There is an option to run the GPP optimization again by clicking ‘Re-Optimize’ button. Clicking ‘Begin RECO’ will begin the RECO optimization function.



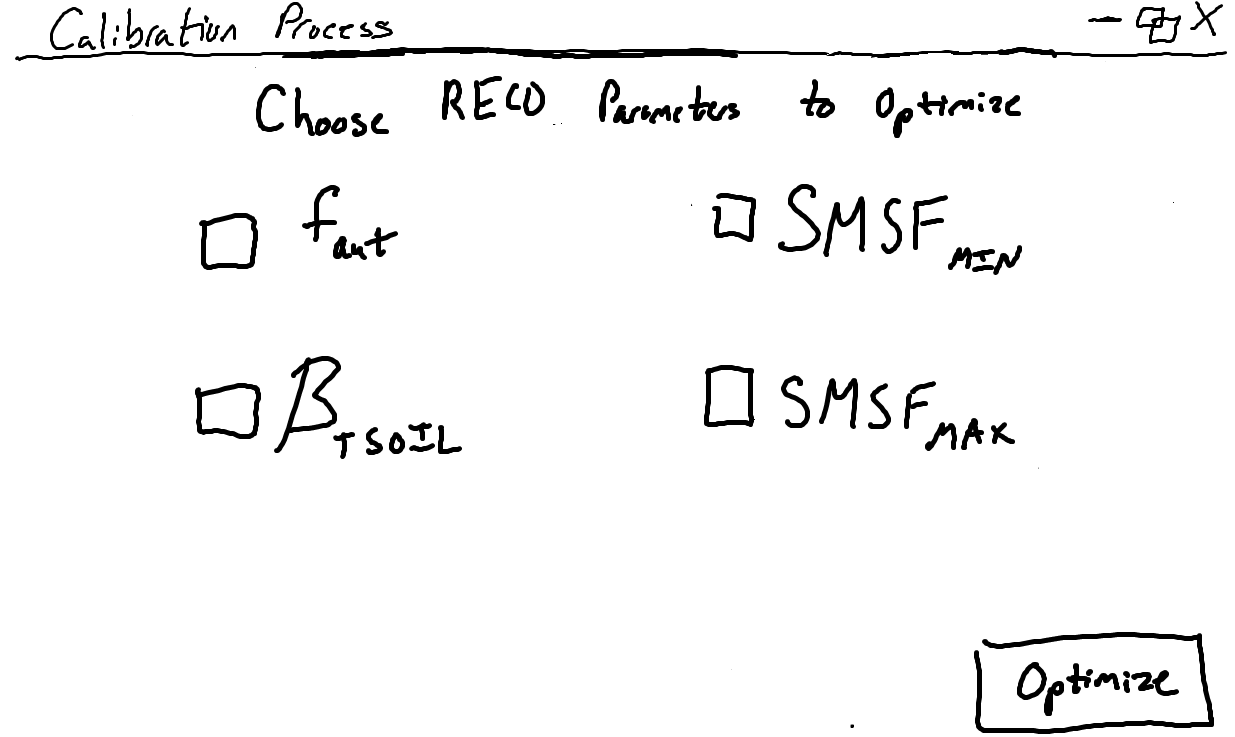
RECO Hyperparameters Page

This page lets the user determine the value, between 0 and 1, of PRH (percentile of RH/Kmult to use in initializing SOC pools) and PK (percentile to use as minimum threshold for acceptable Kmult values). These hyperparameters will be used to calculate Cbar, which is vital to the RECO calculation. Clicking ‘Next’ displays the ramp functions for RECO.



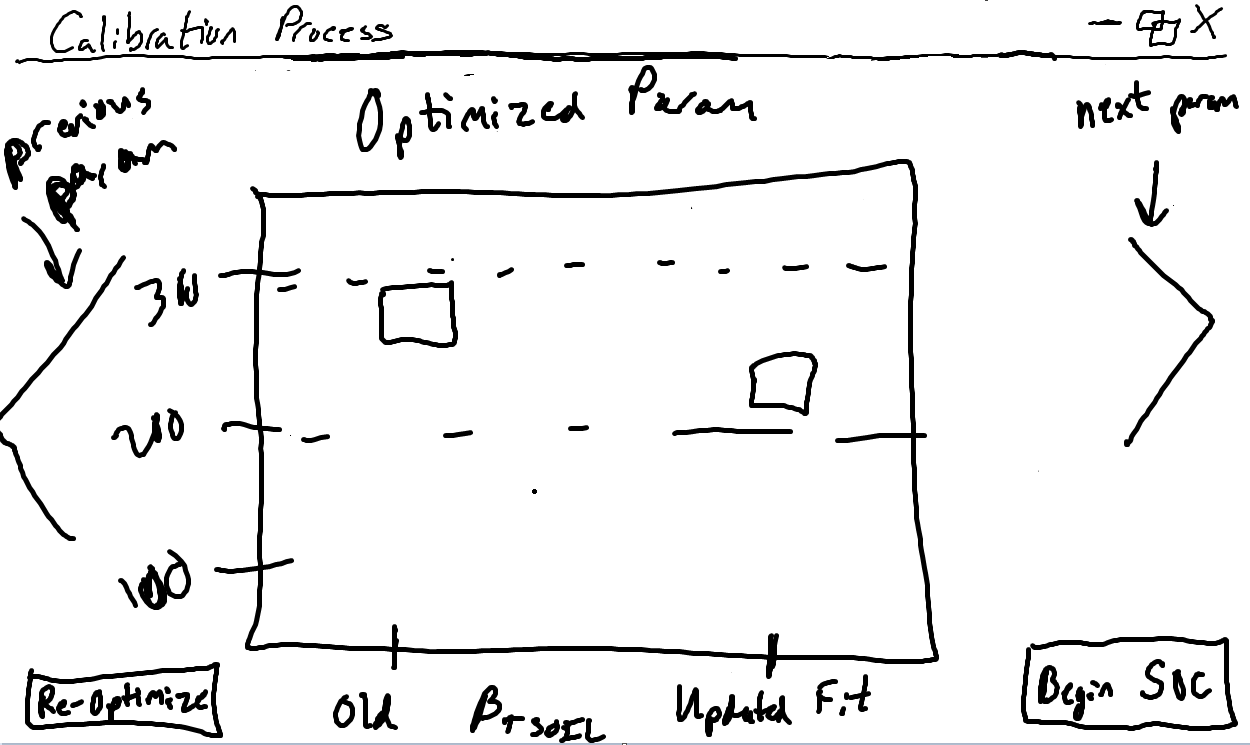
RECO Ramp Function Page

This page displays the plots of the of the current RECO ramp functions, similar to the GPP ramp functions (High priority #7). The user can view two other RECO ramp functions by selecting the arrows on either side of the graph. There is an option to display the graph of RH/Cbar vs Kmult, similar to the GPP vs Emult graph for GPP and showing a page similar to the GPP vs Emult Page with Yes selected when clicking the ‘Next’ button.



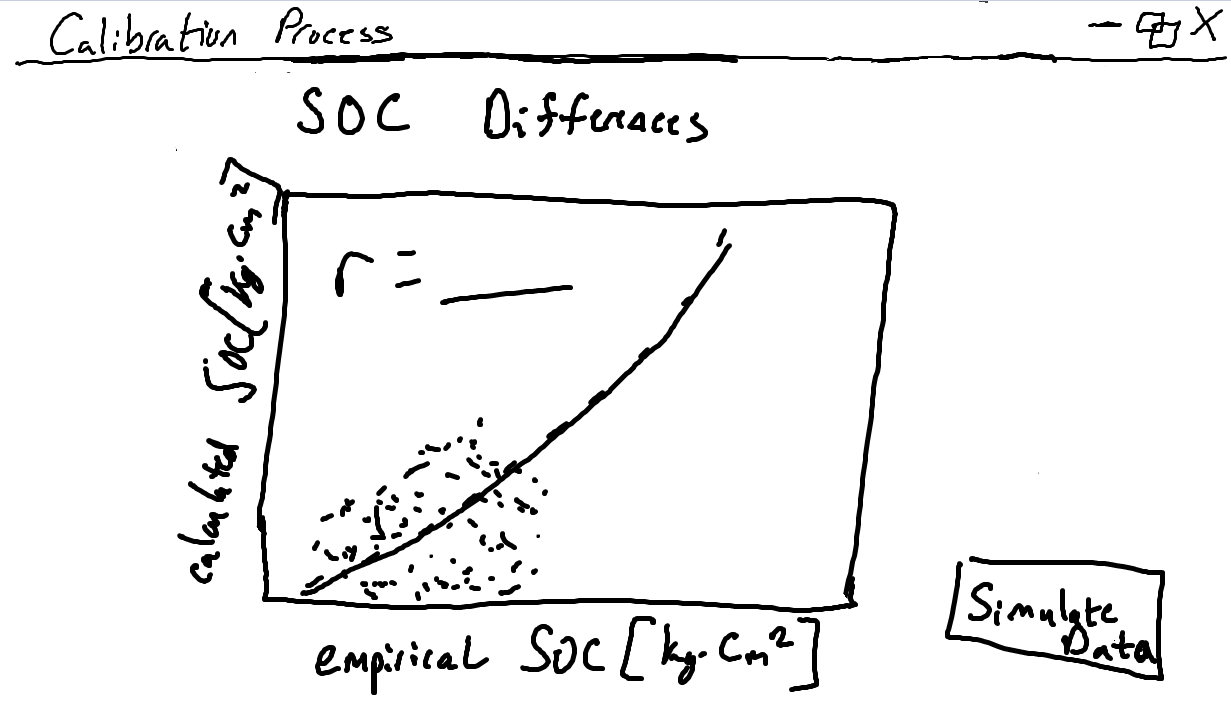
RECO Parameter Page

Much like the GPP Parameter Page, this page allows the user to choose which of the four RECO parameters will be optimized (High priority #6). These four parameters are included as a default but can be unchecked to be excluded. The ‘Optimize’ button begins the RECO optimization process.



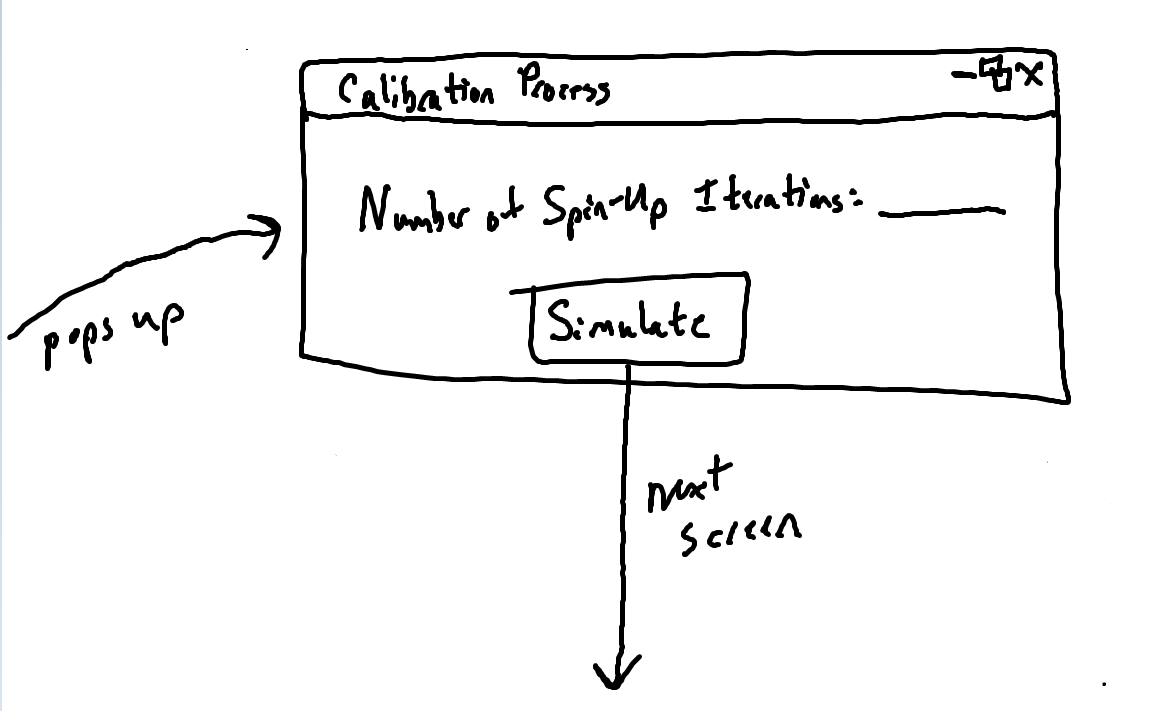
RECO Differences Page

This page displays both the original and updated fit for each optimized parameter, like the GPP Differences page. The value of the updated fit will be reported if it falls on/above/below the upper or lower bound. The updated fit for the optimized RECO parameter is stored and updated in the BPLUT table. There is an option to run the RECO optimization again by clicking ‘Re-Optimize’ button. Clicking ‘Begin SOC’ will begin the final steps to the calibration process.



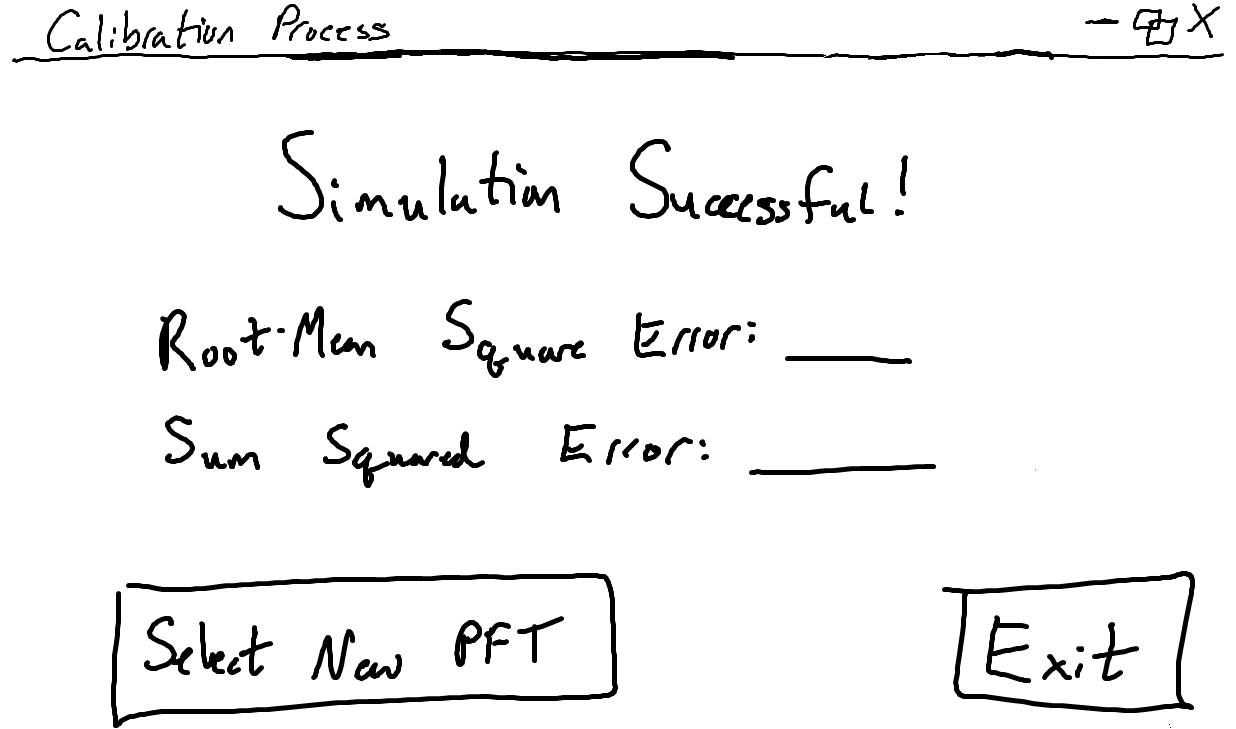
SOC Differences Page

This page displays a plot of the empirical SOC calculated from the ground-truth flux tower data against the model-estimated, or calculated, SOC. The SOC is calculated using the optimized GPP and RECO parameters (High priority #9). The ‘Simulate Data’ button pops up the following window.



Iteration Pop-Up

This pop-up window allows the user to choose the number of numerical spin-up iterations that the model will be doing (High priority #8). The ‘Simulate’ button will begin the simulation iterations.



Ending Page

This page lets the user know that the simulation has run successfully. The root-mean square and sum squared errors are reported to the user for the selected PFT, which is updated to the output. The output of this calibration process, by clicking ‘Exit’, contains the updated BPLUT table, the four SOC stock-size maps, SMRZmin and SMRZmax from 2000-2018 which is also termed the Nature Run (High priority #10). The user can return to the Select PFT Page to continue the calibration process for another PFT by clicking the ‘Select New PFT’ button.

**3. In Class Presentation**

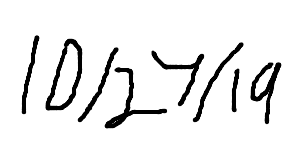
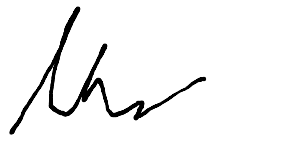
The following list contains the notes and feedback that the team received for this original low-fidelity prototype during the in-class presentation on Wed. Oct 23, 2019:

1. Change the re-smoothing type from a string input to a drop-down menu option in the Selecting Outliers Page
2. Only accept HDF5 file type for input, and throw error when not, on the Configuration File Page
3. For GPP and RECO Parameter Pages, say “upper bound” and “lower bound” instead of “max” and “min”
4. Display definitions and what the value is being calculated with/for when clicking on any term
5. Make back buttons (so the program does not feel strictly linear)
6. Create a ‘Choose/Select All’ button for both Parameter Pages and potentially a ‘Clear All’ to then select one (one needs to be selected to proceed)
7. Create a progress bar, since the calibration process may take a couple hours to run

**4. Execution and Acknowledgement**

The team members hereby indicate by their signatures below that they have read and agree with the specifications of this document.



****

Final Low-fidelity

Prototypes

Document

Prepared by: The Skyentists  
Due Date: 10/24/19

**1. Overview**

UI prototypes take the user through the process of using software without fully implementing the software. UI prototypes consist of rough drafts of the software and have a limited degree of interaction. This is critical to developing an effective UI as it is the most cost and time efficient way to get feedback from users. Developing software is expensive, and it’s even more expensive to re-implement a finished product because the old one had a poor user interface.

There are several different types of prototypes. The first type of prototype implemented in the software development process is a low fidelity prototype. Low fidelity prototypes are typically hand drawn sketches of the software process. These have the advantages of being low cost and easy to edit.

Low fidelity prototypes have a limited degree of interaction, which is why they serve as a benchmark for the other main UI prototype: the high fidelity prototype. High fidelity prototypes are realistic depictions of a program with a high degree of interactivity. High fidelity prototypes let the user give feedback on all parts of the software, especially edge cases. Users can give more meaningful feedback as well, as they can interact with the prototype as if it were the software. These benefits do come at a higher cost than the low fidelity prototype.

Medium level prototypes are a blend between low and high-fidelity prototypes. They are more interactive and expensive than low level prototypes, but not nearly as in depth and costly as high-fidelity prototypes.

**2. Low Fidelity Prototypes**

The high priority functional requirements that are shown by the screenshots below are as followed:

1. Calculate dominant Plant Functional Type (PFT) for each tower site
2. Read in 4 input datasets (L4C (Level 4 Carbon) reference dataset, L4C meteorological input dataset, Flux Tower Fluxes dataset, IGBP FAO (Our empirical one) Soil Organic Carbon Inventory dataset) for calibration
3. Compile the L4C meteorological inputs and flux tower data into a table
4. Allow the user to choose one of the nine PFTs
5. Guide the user through removing outliers in average annual GPP (Gross Primary Production i.e. plants absorbing CO2) and RECO (Ecosystem Respiration Net CO2 output by plants and soil) calculations
6. Allow user to choose which parameters to use when optimizing GPP and RECO
7. Calculate linear ramp functions given current BPLUT (Biome Properties LookUp Table: All the data we have for a single plant type)
8. Allow user to specify number of Numerical Spin-Up iterations and then review results
9. Compute comprehensive validation and fit statistics: graph flux tower data against model-estimated data
10. Output updated BLPUT, 4 SOC (Soil Organic Carbon: CO2 emitted by soil) stock-size maps, and SMRZ (Soil Moisture Root Zone: amount of moisture plants roots can absorb) minimum and maximum for 2000-2018
11. Calculate flux tower weights (some tower sites are in the same 9km section)

A picture containing text

Description automatically generated

Intro Page

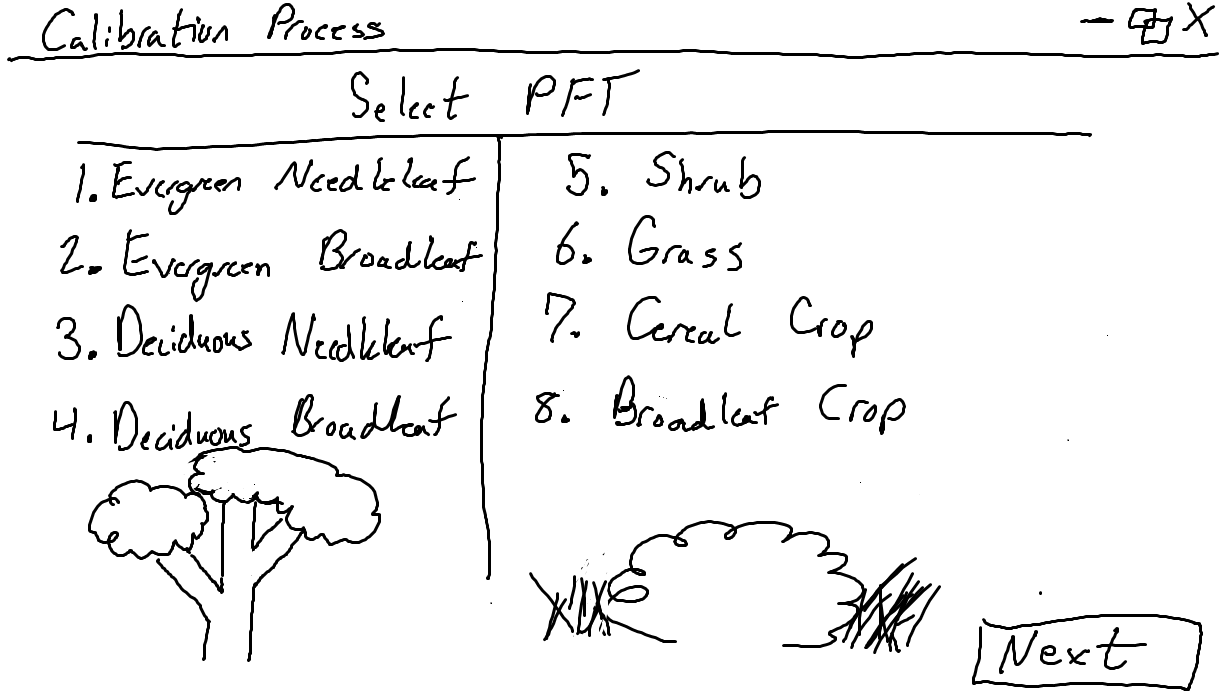
This page shows the first screen of the calibration process. Upon clicking the “Begin Calibration” button, the process of calibrating the SMAP L4C algorithm will start.

A close up of text on a white background

Description automatically generated

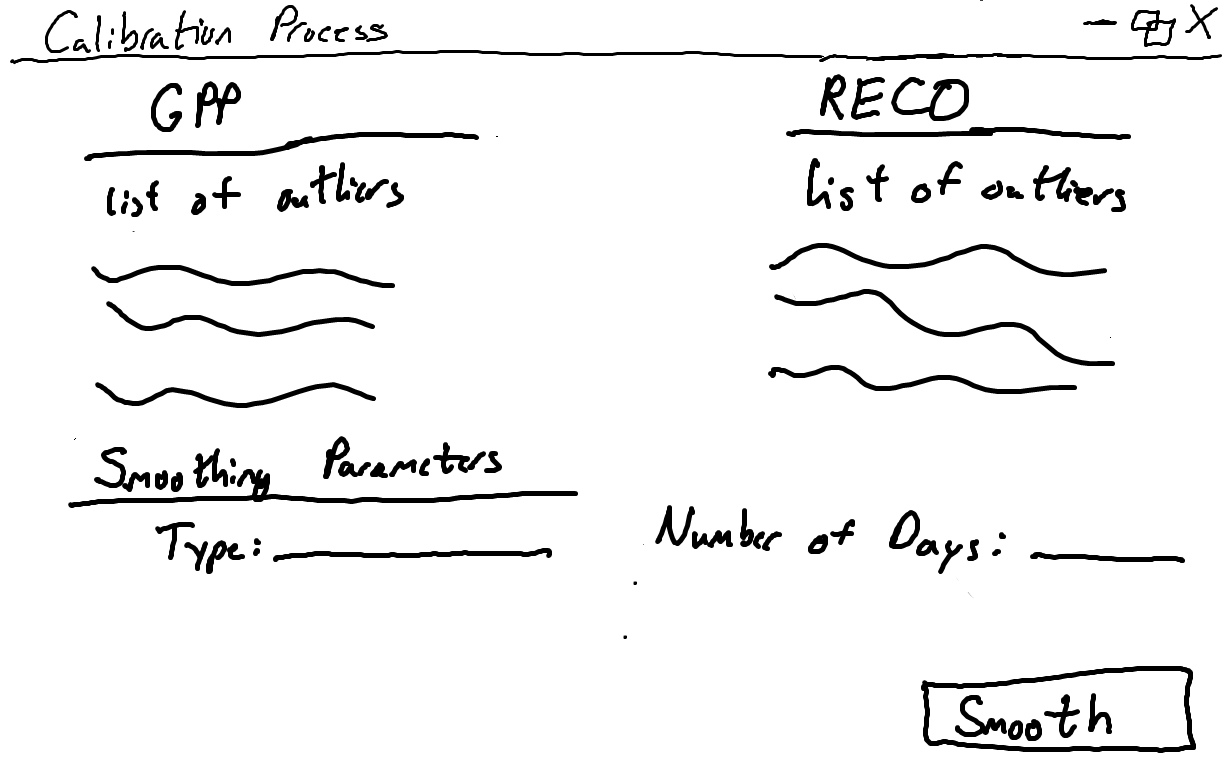
Configuration File Page

This page displays the name of the configuration file that the user will select. The configuration file consists of the current BPLUT table, list of flux towers sites to include and exclude, file path to the previously used Nature Run dataset, file paths for the HDF5 inputs, and the file path for the output file. The ‘Next’ button will allow the reading in of the four input datasets (High priority #2) and the compiling of the meteorological inputs and flux tower data into a single table (High priority #3).



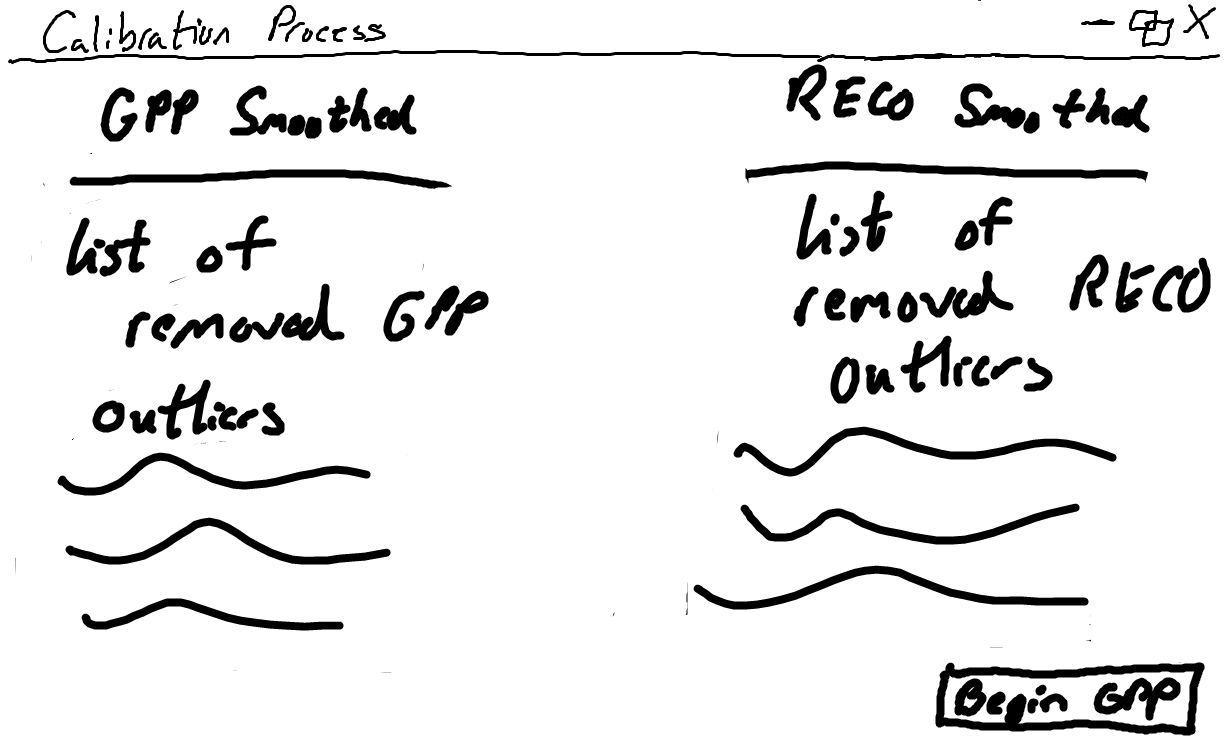
PFT Selection Page

This page lets the user choose 1 of 8 plant functional types (PFTs, high Priority #4). This PFT selection will determine the following ramp function calculations and optimization, as each one will have its own subset of tower data (High priority #1). The selected PFT will be highlighted and the ‘Next’ button will be enabled.



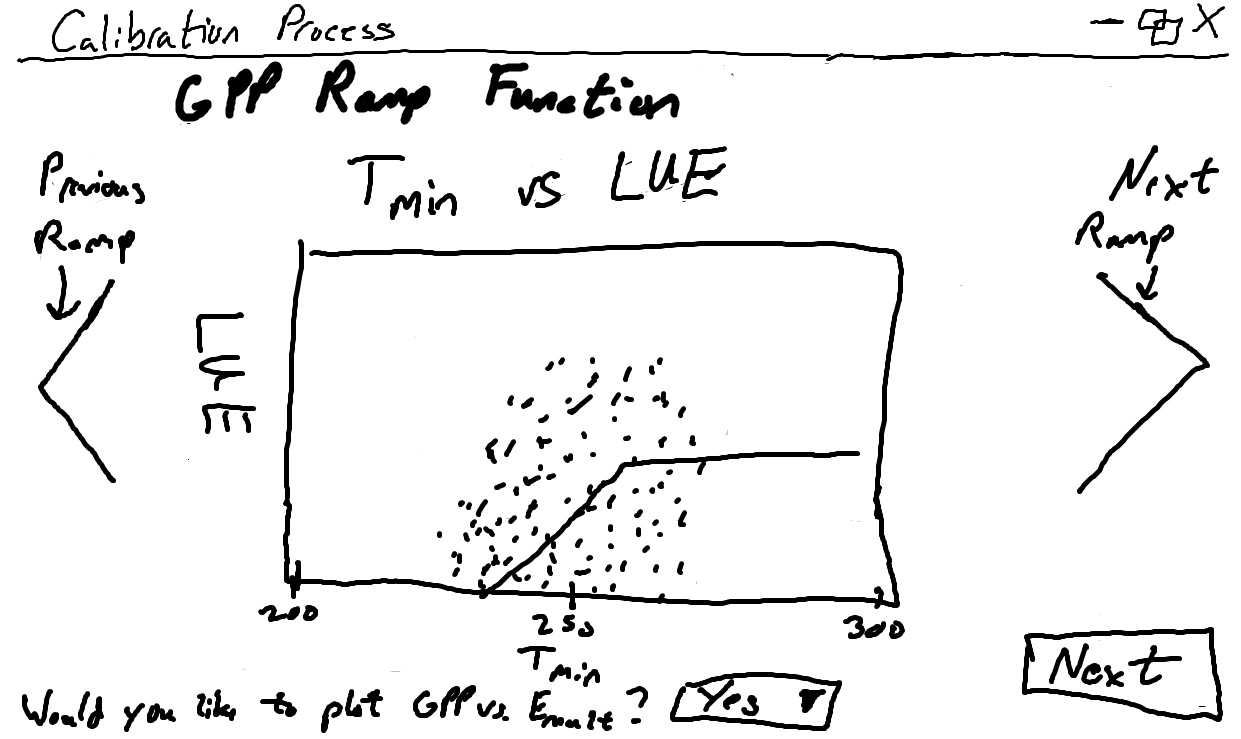
Selecting Outliers Page

This page allows the user to choose outliers in the average annual GPP and RECO that can be removed (High priority #5). The user will also be able to choose the type of smoother to use, such as SciPy’s filtfilt() implementation, and specify the day range of the dataset. Clicking the ‘Smooth’ button will begin the smoothing function.



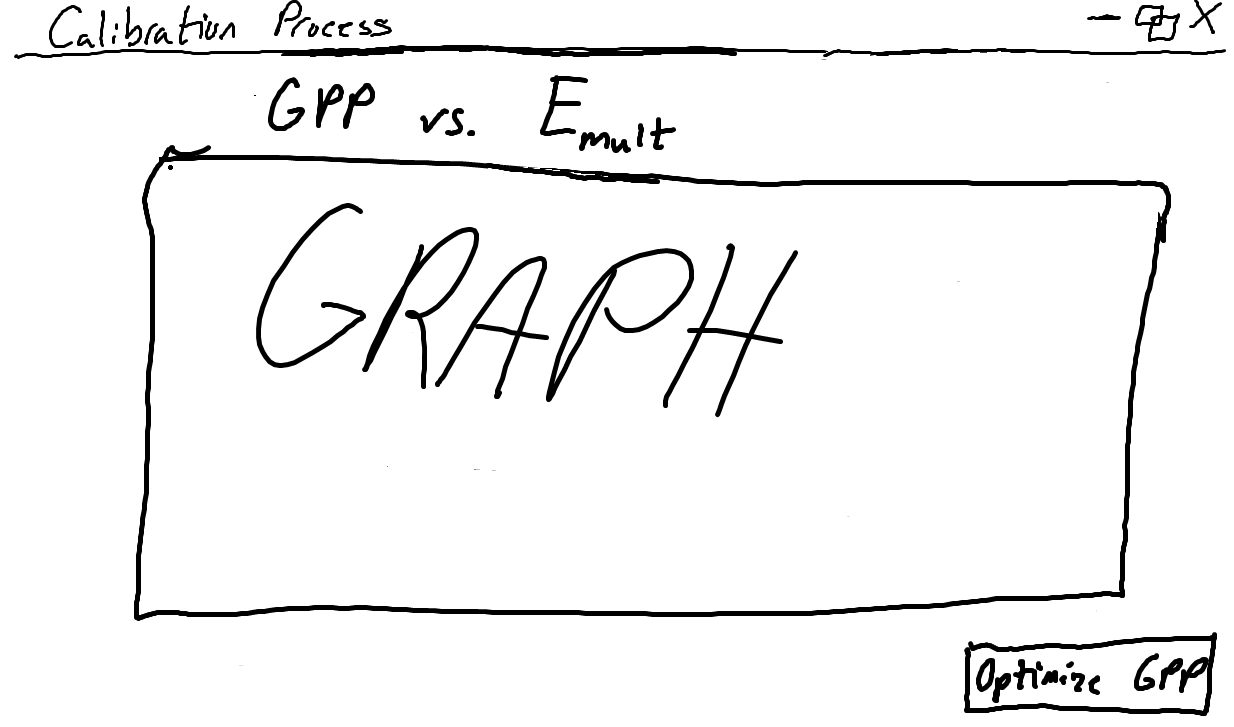
Removed Outliers Page

This page reports the selected outliers that are removed from the subset of data for the selected PFT due to the smoothing function. Clicking ‘Begin GPP’ will start the optimization function.



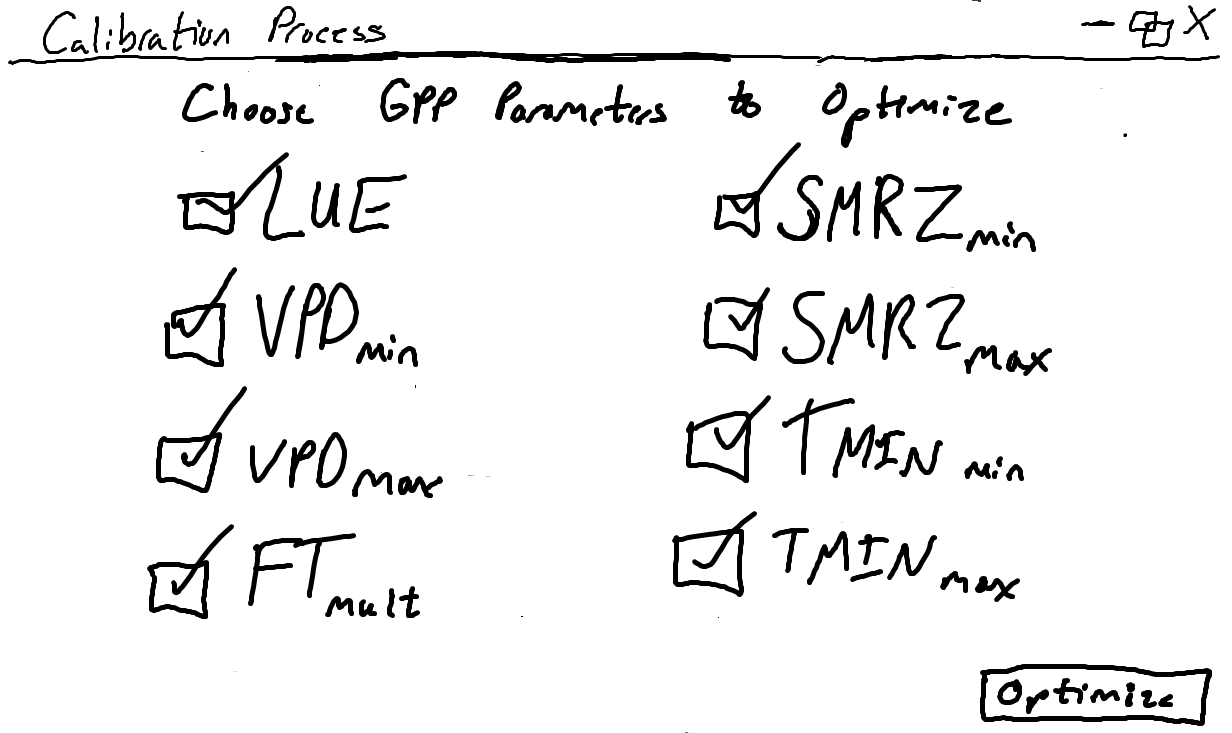
GPP Ramp Function Page

This page displays the plots of the current GPP ramp functions using the current BPLUT table (High priority #7), a part of the configuration file that includes data on the selected PFT. The option to display the GPP vs Emult graph is presented to the user with a dropdown for Yes or No, showing the below page with Yes selected when clicking the ‘Next’ button.



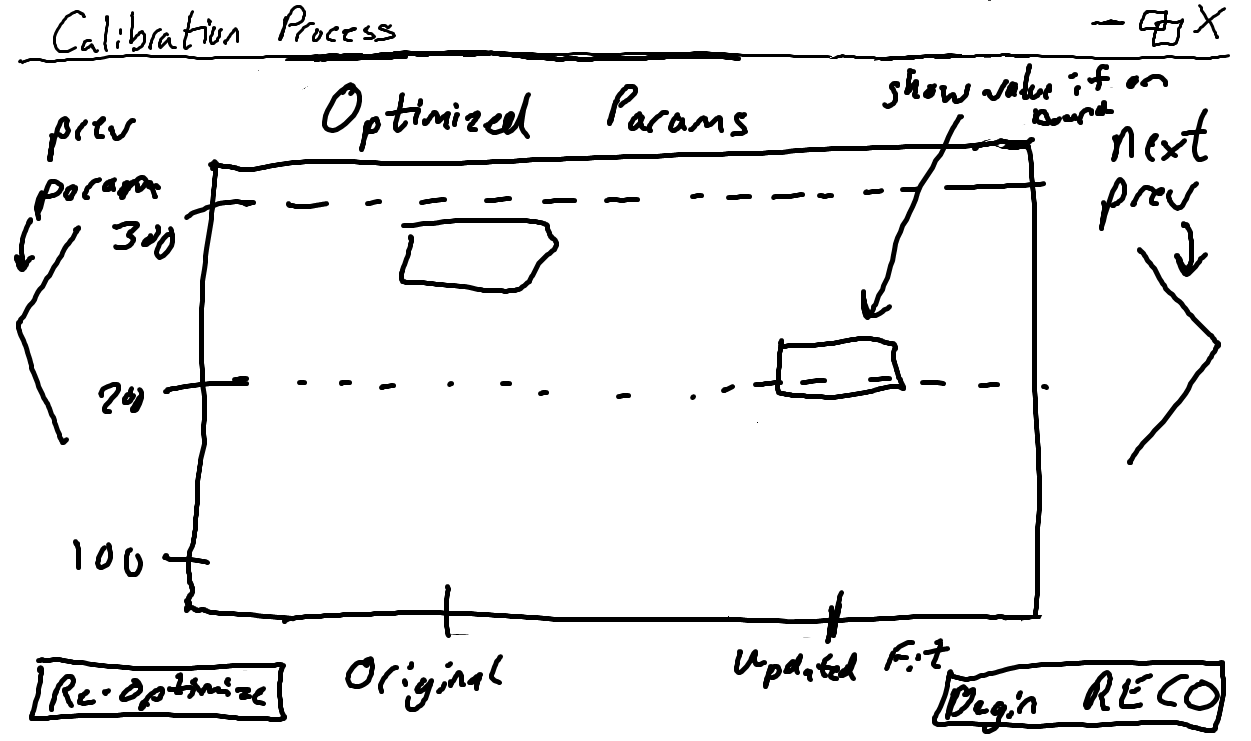
GPP vs Emult Graph Page

This page displays the graph of GPP vs. Emult which is an optional graph that the user may or may not wish to view. Clicking the ‘Optimize GPP’ button will start the optimization of GPP parameters.



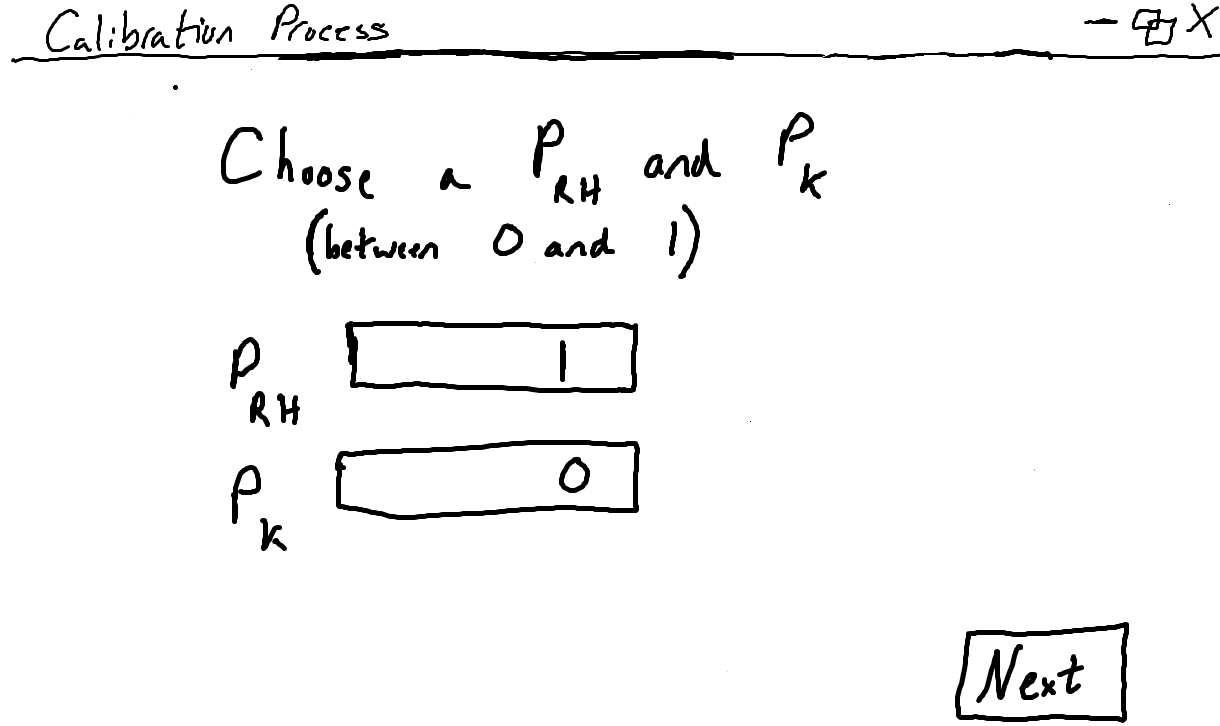
GPP Parameters Page

This page allows the user to choose which of the eight GPP parameters will be optimized (High priority #6). All eight of these parameters will automatically be checked as a default for the optimization process, but can be unchecked to be excluded. The ‘Optimize’ button begins the GPP optimization process.



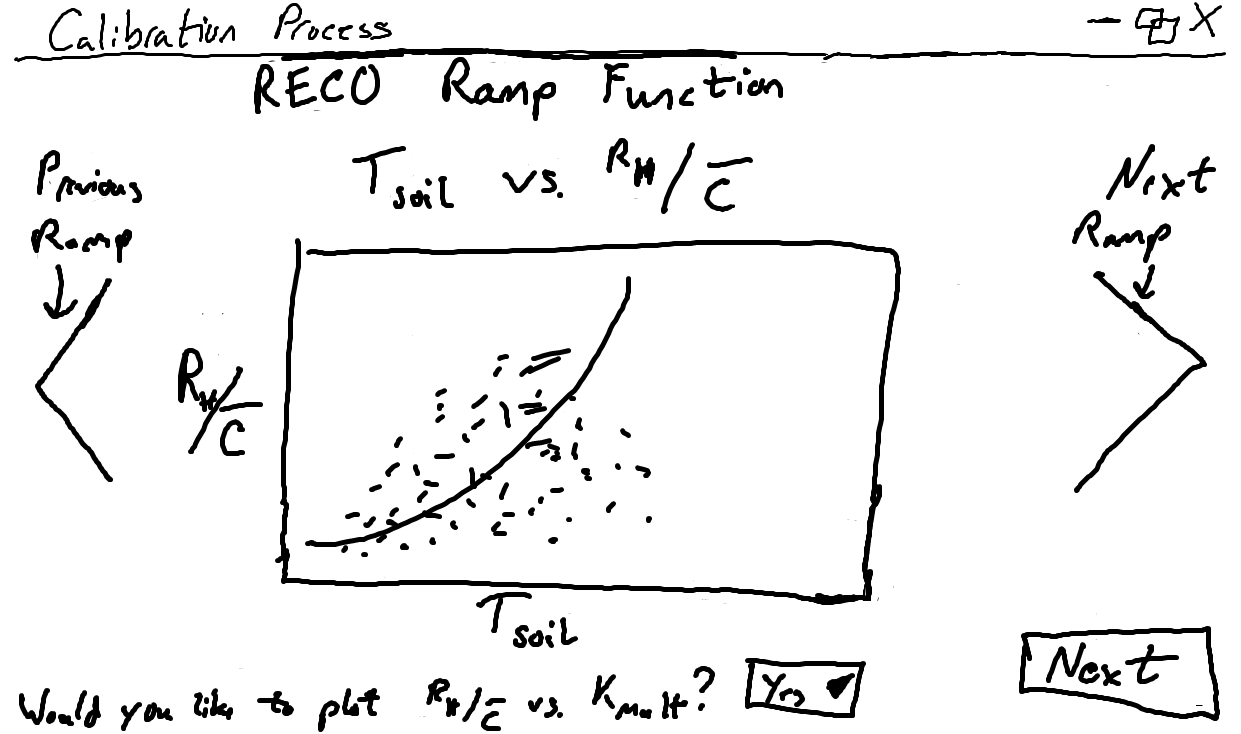
GPP Differences Page

This page displays both the original and updated fit for each optimized parameter. The value of the updated fit will be reported if it falls on or above/below the upper or lower bound. The updated fit is stored and updated in the BPLUT table. There is an option to run the GPP optimization again by clicking ‘Re-Optimize’ button. Clicking ‘Begin RECO’ will begin the RECO optimization function.



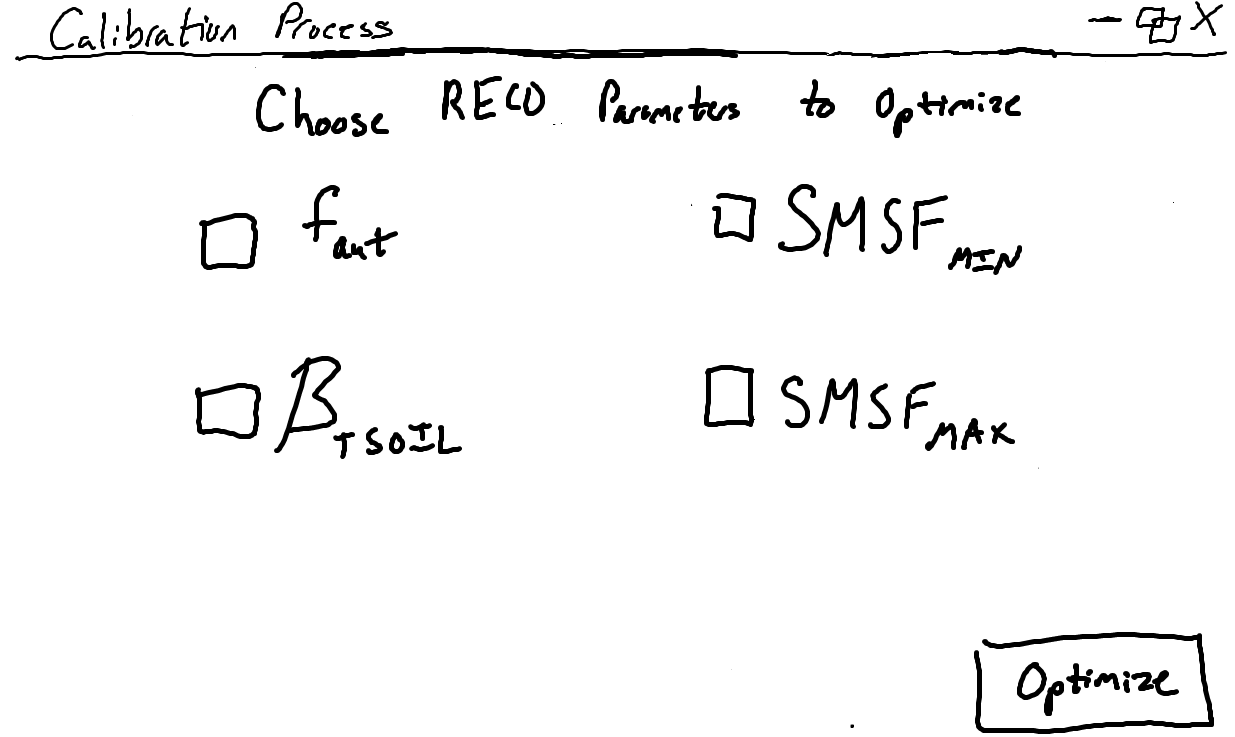
RECO Hyperparameters Page

This page lets the user determine the value, between 0 and 1, of PRH (percentile of RH/Kmult to use in initializing SOC pools) and PK (percentile to use as minimum threshold for acceptable Kmult values). These hyperparameters will be used to calculate Cbar, which is vital to the RECO calculation. Clicking ‘Next’ displays the ramp functions for RECO.



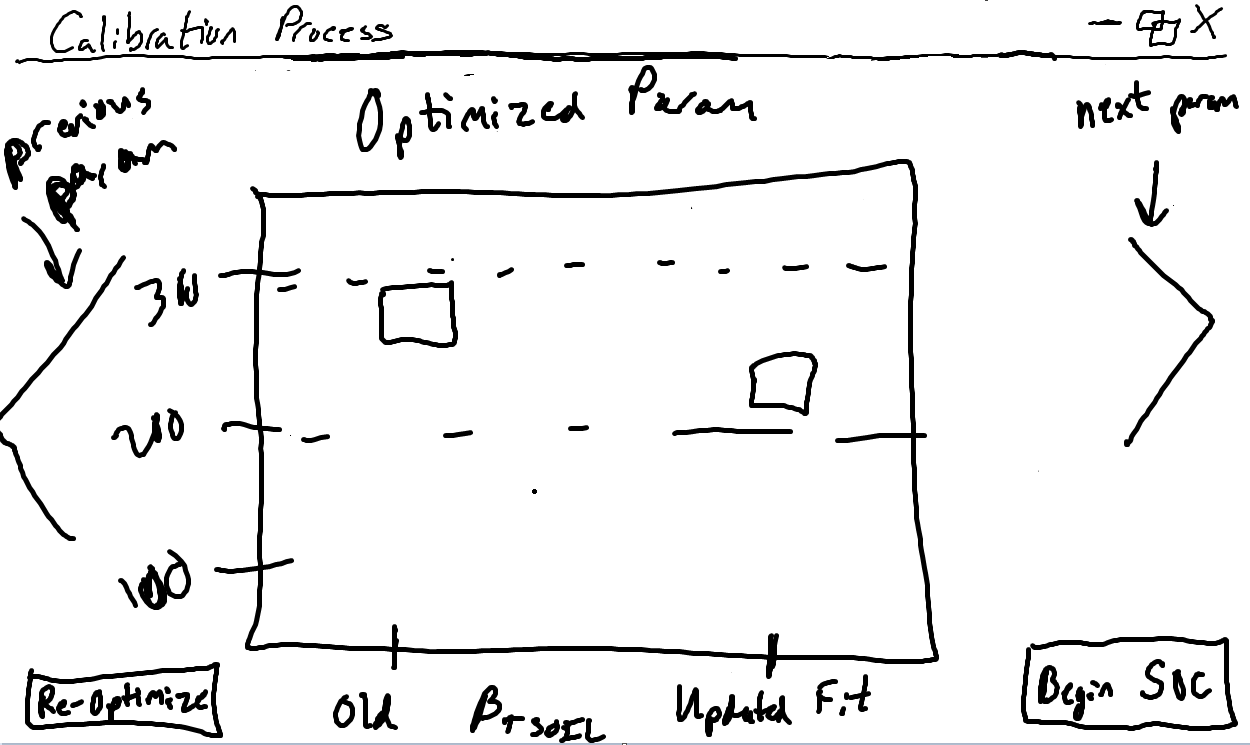
RECO Ramp Function Page

This page displays the plots of the of the current RECO ramp functions, similar to the GPP ramp functions (High priority #7). The user can view two other RECO ramp functions by selecting the arrows on either side of the graph. There is an option to display the graph of RH/Cbar vs Kmult, similar to the GPP vs Emult graph for GPP and showing a page similar to the GPP vs Emult Page with Yes selected when clicking the ‘Next’ button.



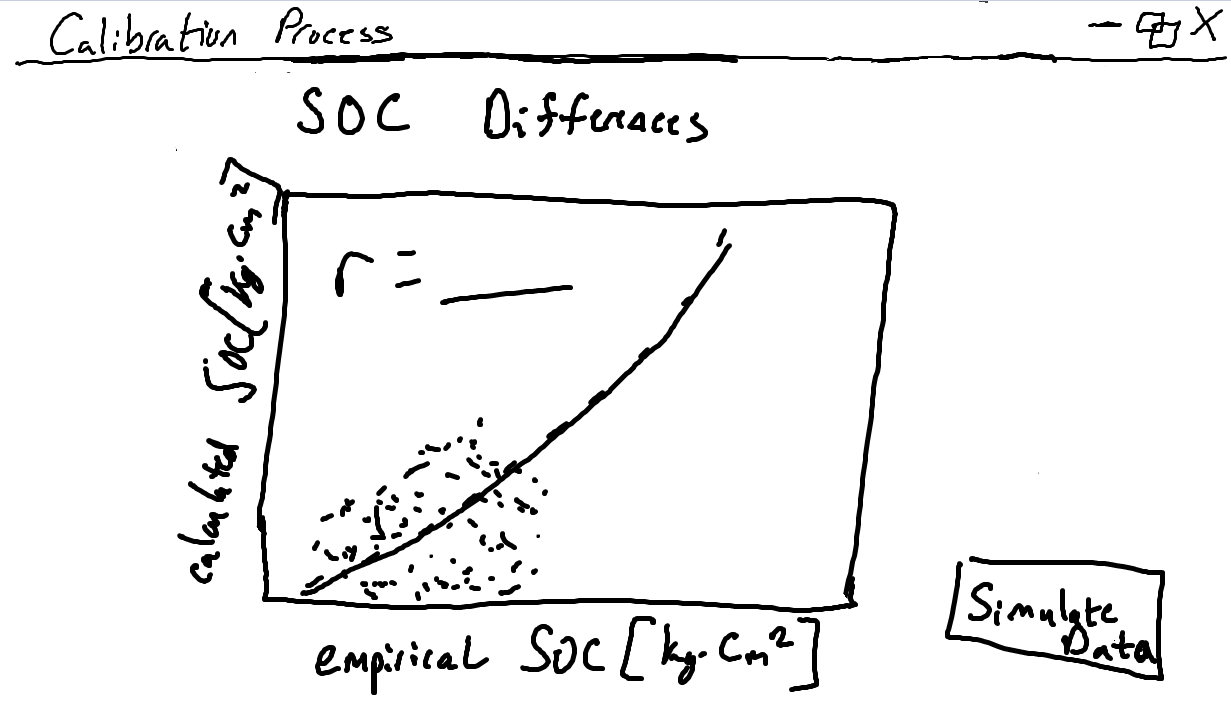
RECO Parameter Page

Much like the GPP Parameter Page, this page allows the user to choose which of the four RECO parameters will be optimized (High priority #6). These four parameters are included as a default but can be unchecked to be excluded. The ‘Optimize’ button begins the RECO optimization process.



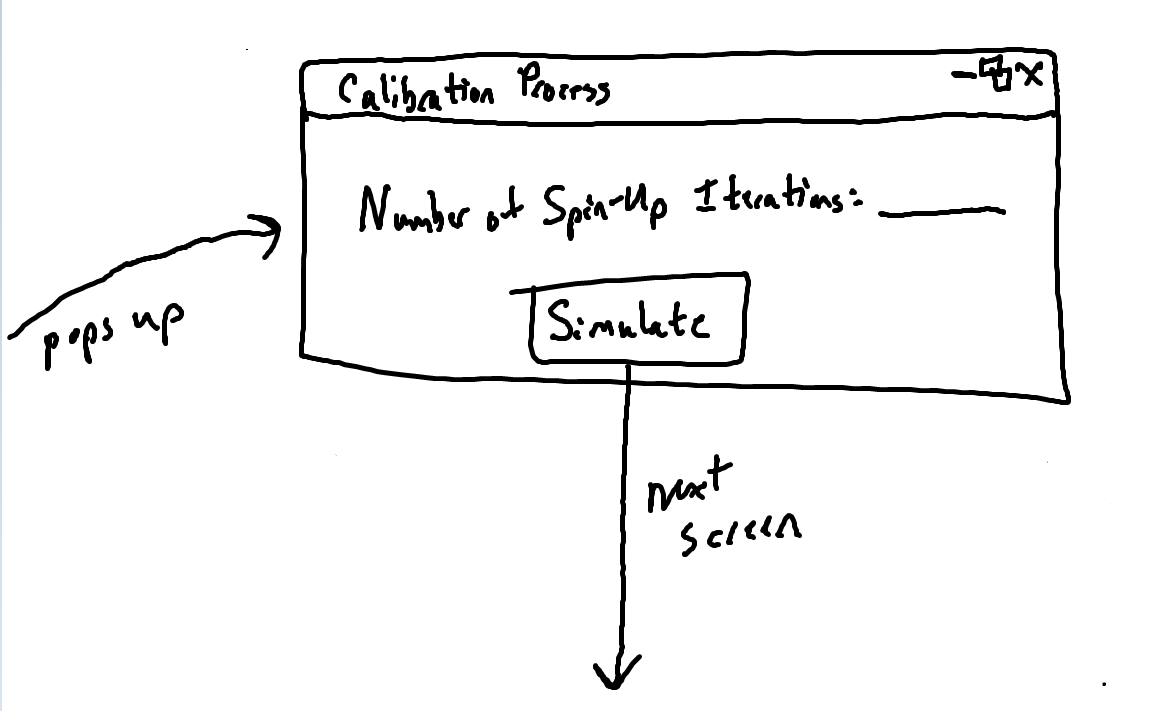
RECO Differences Page

This page displays both the original and updated fit for each optimized parameter, like the GPP Differences page. The value of the updated fit will be reported if it falls on/above/below the upper or lower bound. The updated fit for the optimized RECO parameter is stored and updated in the BPLUT table. There is an option to run the RECO optimization again by clicking ‘Re-Optimize’ button. Clicking ‘Begin SOC’ will begin the final steps to the calibration process.



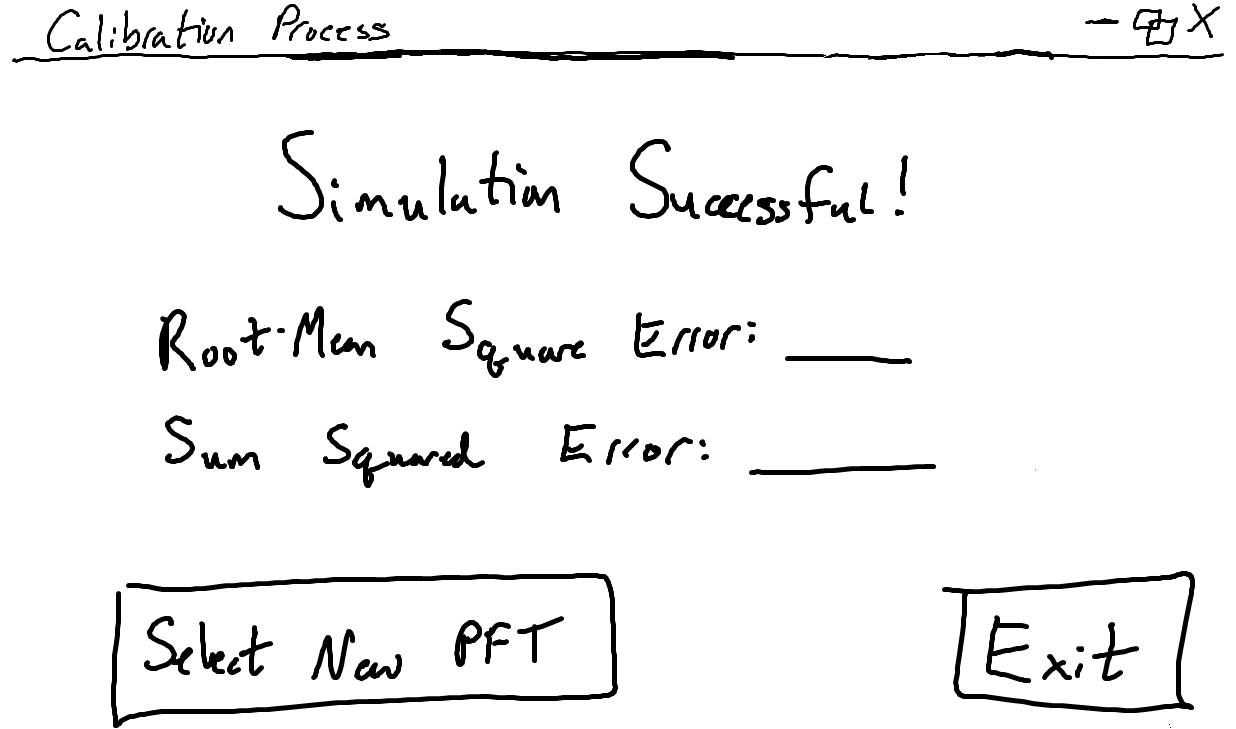
SOC Differences Page

This page displays a plot of the empirical SOC calculated from the ground-truth flux tower data against the model-estimated, or calculated, SOC. The SOC is calculated using the optimized GPP and RECO parameters (High priority #9). The ‘Simulate Data’ button pops up the following window.



Iteration Pop-Up

This pop-up window allows the user to choose the number of numerical spin-up iterations that the model will be doing (High priority #8). The ‘Simulate’ button will begin the simulation iterations.



Ending Page

This page lets the user know that the simulation has run successfully. The root-mean square and sum squared errors are reported to the user for the selected PFT, which is updated to the output. The output of this calibration process, by clicking ‘Exit’, contains the updated BPLUT table, the four SOC stock-size maps, SMRZmin and SMRZmax from 2000-2018 which is also termed the Nature Run (High priority #10). The user can return to the Select PFT Page to continue the calibration process for another PFT by clicking the ‘Select New PFT’ button.

**3. In Class Presentation**

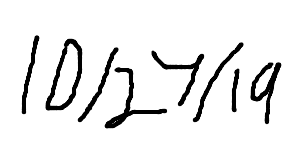
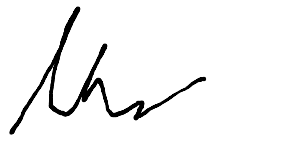
The following list contains the notes and feedback that the team received for this original low-fidelity prototype during the in-class presentation on Wed. Oct 23, 2019:

1. Change the re-smoothing type from a string input to a drop-down menu option in the Selecting Outliers Page
2. Only accept HDF5 file type for input, and throw error when not, on the Configuration File Page
3. For GPP and RECO Parameter Pages, say “upper bound” and “lower bound” instead of “max” and “min”
4. Display definitions and what the value is being calculated with/for when clicking on any term
5. Make back buttons (so the program does not feel strictly linear)
6. Create a ‘Choose/Select All’ button for both Parameter Pages and potentially a ‘Clear All’ to then select one (one needs to be selected to proceed)
7. Create a progress bar, since the calibration process may take a couple hours to run

**4. Execution and Acknowledgement**

The team members hereby indicate by their signatures below that they have read and agree with the specifications of this document.





**Old User Feedback for UI Prototype**

General (All Pages):

1. Class:
   1. Make back buttons (so the program does not feel strictly linear)
   2. Display definitions and what the value is being calculated with/for when clicking on any term
2. Professor Reimer
   1. Back OR exit options

Read in Config File Page (Page 4):

1. Class:
   * Only accept HDF5 file type for input, and throw error when not, on the Configuration File Page
2. Professor Reimer:
   * Have a place to write the output filename in ‘Read in Config’ page

Plant Functional Type (PFT) Selection Page (Page 5):

1. Client:
   * PFTs used in calibration should be indicated in the configuration file and/or the structure of the input datasets (e.g., flux tower dataset will have a dominant PFT column and PFT-by-share-of-1km-pixels columns).
2. Professor Reimer:
   * Dynamically change image for which PFT is selected

Selecting Outliers from a List and Display Subset of Smoothed Outliers Pages (Pages 6-7):

1. Client:
   * Instead of listing outliers for GPP and RECO, it would be more informative to show histograms of the distributions before and after outlier removal, along with the total number of outliers removed. Changing "Number of Days" to "Window Size" will also improve generality.
2. Class:
   * Change the re-smoothing type from a string input to a drop-down menu option in the Selecting Outliers Page
3. Professor Reimer:
   * Clicking outliers to remove them in ‘Outliers’ Page or a box to include/exclude
   * Add a choose which \_\_\_ outliers to include/ignore
   * Include log of previously removed outliers and report (potential notes for client)

Choosing Parameters to Optimize GPP and RECO Pages (Page 10):

* Class:
  + Create a ‘Choose/Select All’ button for both Parameter Pages and potentially a ‘Clear All’ to then select one (one needs to be selected to proceed)
  + For GPP and RECO Parameter Pages, say “upper bound” and “lower bound” instead of “max” and “min”

Choosing Hyperparameters for RECO optimization Page (Page 12):

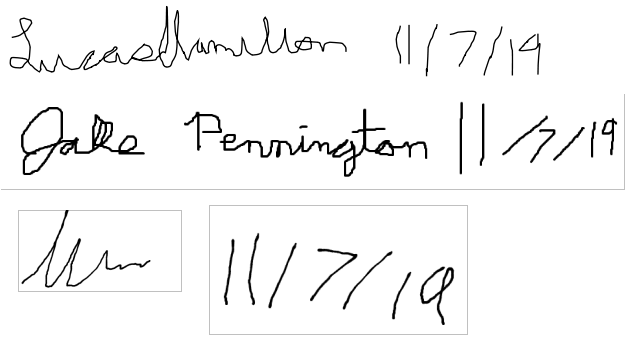
* Professor Reimer:
  + Put a slider bar and text box for Prh and Pk

Pages to be added:

* Client:
  + Ending Page will show fit statistics such as Pearson’s r value and RMSE for GPP, RECO, and NEE (see updated document from client)
* Class:
  + Create a progress bar, since the calibration process may take a couple hours to run

Signatures





**Final User Feedback for UI Prototype**

**Overview**

This document provides the categorization of the user interface prototype feedback provided by peers and others. User feedback is necessary for the design of a prototype, because it will help provide the developers with ideas on how to efficiently code the project. This feedback also lets the developer get a general feel on how users will navigate through the interface and the specific way the client wants the program to look. Feedback is also useful to ensure that the developers and client are on the same page after gathering requirements and previous documentation.

**Feedback List**

This is the list of feedback received from the users of our project. The list is categorized by screenshots of the low-fidelity user interface design, described as “pages”. The list is then sub-categorized by users, which include the client at NTSG, classmates/peers, and the professor of the class. The team is planning on implementing most of the suggested feedback into the project, because the feedback will allow for a smoother user experience during the calibration process. The only feedback that may not be implemented would be the “Pages to be added” section since there may be time constraints for coding the user interface.

General (All Pages):

* Class:
  + Make back buttons (so the program does not feel strictly linear)
  + Display definitions and what the value is being calculated with/for when clicking on any term
* Professor Reimer:
  + Back OR exit options

Read in Config File Page (Page 4):

* Class:
  1. Only accept HDF5 file type for input, and throw error when not, on the Configuration File Page
* Professor Reimer:
  1. Have a place to write the output filename in ‘Read in Config’ page

Plant Functional Type (PFT) Selection Page (Page 5):

1. Client:
   * PFTs used in calibration should be indicated in the configuration file and/or the structure of the input datasets (e.g., flux tower dataset will have a dominant PFT column and PFT-by-share-of-1km-pixels columns).
2. Professor Reimer:
   * Dynamically change image for which PFT is selected

Selecting Outliers from a List and Display Subset of Smoothed Outliers Pages (Pages 6-7):

* Client:
  1. Instead of listing outliers for GPP and RECO, it would be more informative to show histograms of the distributions before and after outlier removal, along with the total number of outliers removed. Changing "Number of Days" to "Window Size" will also improve generality.
* Class:
  1. Change the re-smoothing type from a string input to a drop-down menu option in the Selecting Outliers Page
* Professor Reimer:
  1. Clicking outliers to remove them in ‘Outliers’ Page or a box to include/exclude
  2. Add a choose which GPP/RECO outliers to include or ignore
  3. Include log of previously removed outliers and report (potential notes for client)

Choosing Parameters to Optimize GPP and RECO Pages (Page 10):

* Class:
  + Create a ‘Choose/Select All’ button for both Parameter Pages and potentially a ‘Clear All’ to then select one (one needs to be selected to proceed)
  + For GPP and RECO Parameter Pages, say “upper bound” and “lower bound” instead of “max” and “min”

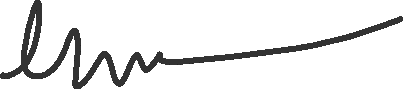
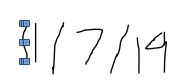
Choosing Hyperparameters for RECO optimization Page (Page 12):

* Professor Reimer:
  + Put a slider bar and text box for Prh and Pk

Pages to be added:

* Client:
  + Ending Page will show fit statistics such as Pearson’s r value and RMSE for GPP, RECO, and NEE (see updated document from client)
* Class:
  + Create a progress bar, since the calibration process may take a couple hours to run

**Acknowledgement**

The team members hereby indicate by their signatures below that they have read and agree with the specifications of this document



**Implementation Plan CSCI 426**

**Group Name: \_\_\_The Skyentists\_\_\_ Fall 2019**

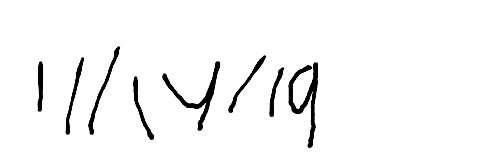
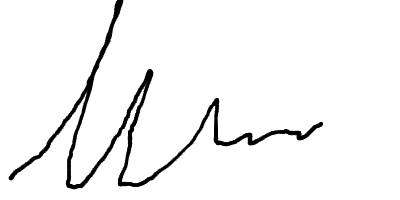
|  |  |  |  |
| --- | --- | --- | --- |
| **Pre- implementation tasks\*** | **Who is responsible** | **Time estimate to complete (in hours)** | **Targeted start date and date of completion** |
| **Initialize Github repository**  **-Client will be owner of repo** | **Max** | **.5 hours** | **11/4 to 12/1** |
| **Study Requirements document from client (compare with sys docs)** | **Team** | **20 hours per person (can be higher if needed)** | **11/4 to 12/31** |
| **Acquire sample datasets from client to read in** | **Mark** | **1 hour** | **11/18 to 12/1** |
| **Install Python 3 and PyQT (for GUI)** | **Team** | **1 hour per person** | **12/1 to 12/31** |
| **Study Python/PyQT to ensure familiarity** | **Team** | **5 hours per person** | **12/1 to 12/31** |
|  |  |  |  |
|  |  |  |  |

Due to the nature of our project, there are not a lot of pre-implementation tasks that need to be completed. The most vital task is each group member familiarizing themselves with the document provided by the client.

Complete a similar table for the start of the implementation phase. This will be a working document that we re-visit on the first day of the spring semester.

|  |  |  |  |
| --- | --- | --- | --- |
| **Major implementation tasks\*** | **Who is responsible** | **Time estimate to complete (in hours)** | **Targeted start date and date of completion** |
| **Generate base classes for similar GPP and RECO processes** | **Lucas** | **7 hours** | **1/1 to 1/20** |
| **Create calibration process graphs with PyQT** | **Max** | **10 hours** | **1/15 to 2/15** |
| **Proper design of the configuration file and file tree for datasets** | **Mark** | **6 hours** | **1/1 to 1/7** |
| **Construct in-memory BPLUT table to compare against input BPLUT** | **Jake** | **3 hours** | **1/1 to 1/7** |
| **Ensure that the program will be able to be run locally (no web servers or databases)** | **Team** | **.5 hours** | **3/1 to 3/7** |
| **Create special functions for each calculation that is not included in GPP and RECO (SOC functions and GUI)** | **Team** | **15 hours** | **1/1 to 2/1** |
| **Optimization of functions to ensure a fast calibration process** | **Jake** | **10 hours** | **3/1 to 4/1** |
| **Style the GUI** | **Mark** | **4 hours** | **3/1 to 3/7** |
|  |  |  |  |

**Acknowledgement**

The team members hereby indicate by their signatures below that they have read and agree with the specifications of this document.A close up of a logo

Description automatically generated

High-Fidelity

UI Prototype

Prepared by: The Skyentists  
Due Date: 12/05/19

**1. Overivew**

The high-fidelity prototype is the interactive representation between the user and the software that is closest to the final design and implementation. These prototypes let users give feedback on all parts of the software, especially when dealing with edge cases. Users can give more meaningful feedback than lower-fidelity prototypes, as they can interact with the prototype as if it were the software. The users can see how the interface will function and behave, while showing off what is available for viewing during the software’s runtime.

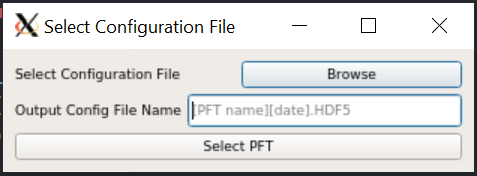
**2. High Fidelity Prototype**

The full high-fidelity prototype for the calibration software can be found in this git repository: <https://github.com/markm700/CalibrationUIPrototype>

This is a public repository that is cloned from the private repository that the client created (and is private) that includes the project documents and medium to high fidelity prototype. Input the command “python3 calibration.py” in the folder that the prototype is in to begin the calibration process.

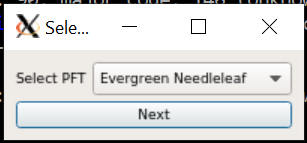
Apart from viewing the full prototype, this section highlights the changes made from the low-fidelity prototype to creating this functioning prototype. Most of these changes were based on the original user feedback from all user groups (the class, client, and professor), although some decisions were made by the team to change the prototype. The team decided not to include the functionality of a back button, as this calibration process is very linear and needs to be followed in that order. However, there is the exit option at any time to stop the calibration process by clicking the exit button in the top right corner. The addition of a pop-up “help” menu when clicking on a definition was also not implemented but will be added later.

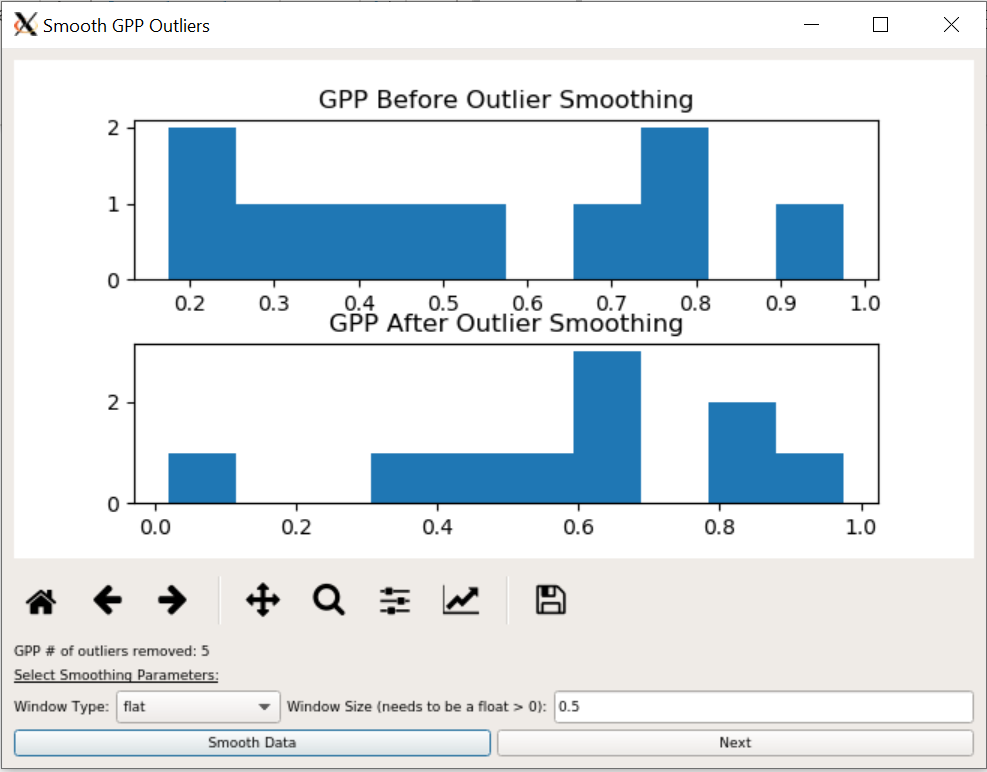
Configuration File Page



This page now gives the user an option to name the output configuration file while choosing the current config file. The output file naming was a function that was suggested in the feedback. This functionality happens here because the output configuration file will be written once all the plant functional types (PFT) for that file have been calibrated.

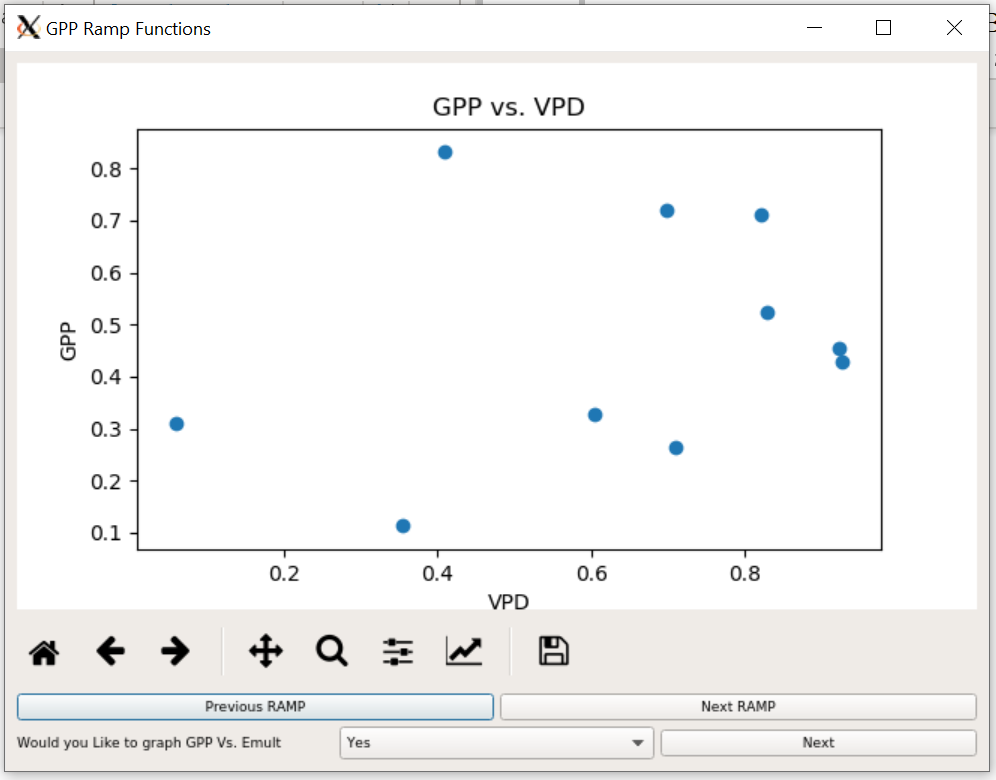
PFT Selection Page

  
The changes to this page were the most drastic from the low-fidelity prototype. Based on feedback from all sources, the plant functional types (PFTs) will be read in from the current configuration file and will not be strictly limited to the original 8 PFTs. Pictures were not included in this prototype but can be added later depending on feedback.

Selecting Outliers Page

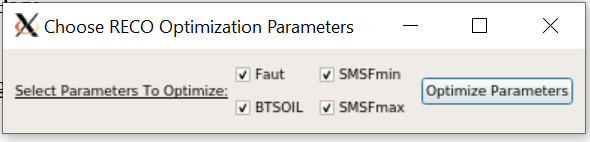
The changes to this page from the low-fidelity prototype were that the GPP and RECO outliers both have their own distinct page in this prototype. This change was based upon client feedback, as histograms would be easier to view the window size and outliers removed. The number of outliers removed for that process are now shown.

Ramp Function Pages



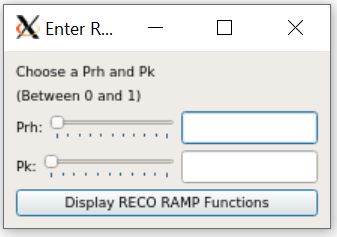
This page still displays the ramp functions as in the low fidelity prototype. The previous and next ramp buttons were changed from arrows to buttons for easy identification in this prototype, although the team and client are open to changing this.

Parameter Pages



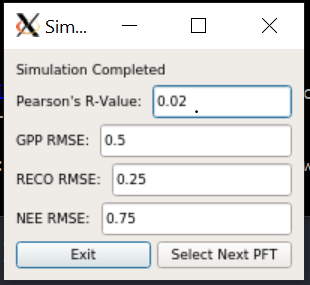
This page displays all the parameters for RECO (or GPP) that are being chosen to optimize. These parameters are now all included, by default, to be optimized. Although the class provided feedback to select/de-select all parameters, the team did not decide to implement this since a majority of or all parameters will be optimized in most cases. Only rarely will one or two parameters be selected.

RECO Hyperparameters Page



The change to this page was the addition of a slider bar. If the value (a float between 0 and 1) is changed by the slider or text field, then the value of the other one will update with it. The user inputting a number greater than 1 will cause the slider of either Prh or Pk to disappear and an error message to be thrown before proceeding.

Ending Page



The changes to this ending page include the root mean square error (RMSE) of GPP, RECO, and NEE as well as Pearson’s R-value. This decision was made based upon client feedback. The values of these fields are read only.

**3. User Feedback**

Feedback on this prototype was provided by peers (11 total) and the client (only 1) prior to the final presentation. The class provided some feedback during the presentation. All the feedback provided on the high-fidelity prototype is categorized by page and then further divided by the user groups

All Pages/General:

Peers:

* + Get rid of taskbars (may be too many clicks/buttons for user)
  + Better scaling of y-axis for graphs
  + Make graph titles uniform (x vs y OR y vs x, not both)
  + Make fonts a little bigger for easier readability
  + Window resizing too small for some pages

Client:

* + Include indicator of which PFT is currently being calibrated
  + Include some sort of progress indication for that specific PFT
  + Window resizing is cutting off the dialog titles

Class:

* + Start over/exit button
  + Fix resizing of windows (encompass in a larger shell)
  + Have option to view just the current values of the BPLUT table
  + Keep track of calibration steps and current PFT calibrated
  + Make tooltips more easily accessible

Outliers Page:

Peers:

* + Placeholder “Required” text in Window Size fields

Ramp Function Pages:

Peers:

* + Change default from yes to no to make the optional graph

Client:

* + Display the old and new values of the optimized parameters next to RAMP functions (Example: old and new VPDmin and VPDmax listed next to VPD ramp)

Class:

* + Provide axis labels for all graphs
  + Make graphs side-by-side or switch the next/previous ramp buttons to a tab

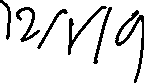
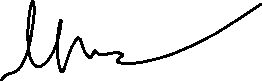
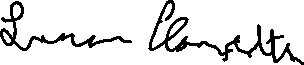
SOC Plot Page:

Peers:

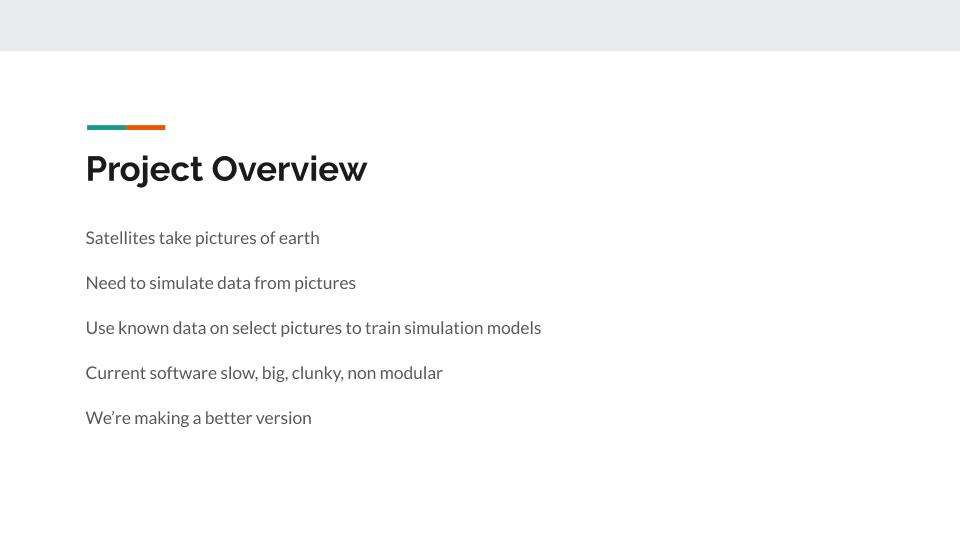
* + Remove underscores from SOC

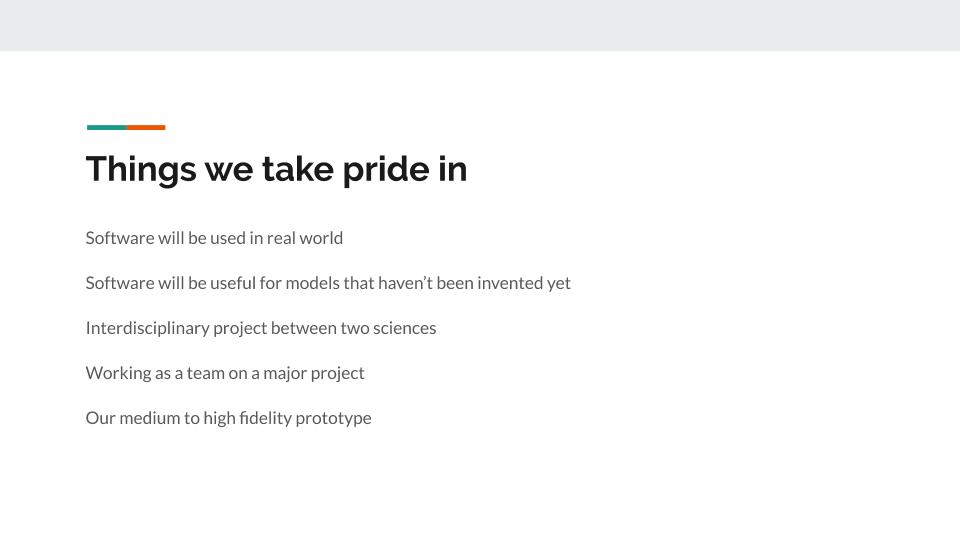
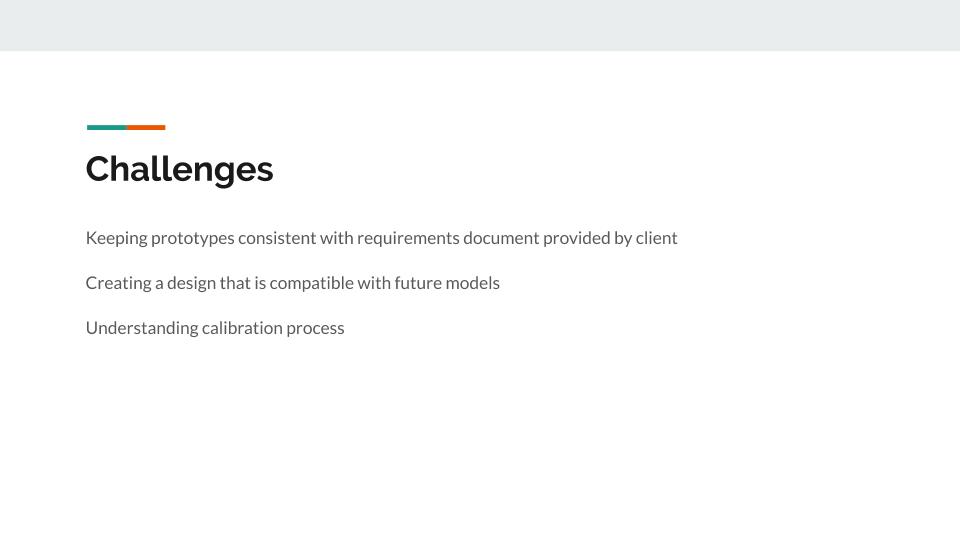
**4. Execution and Acknowledgement**

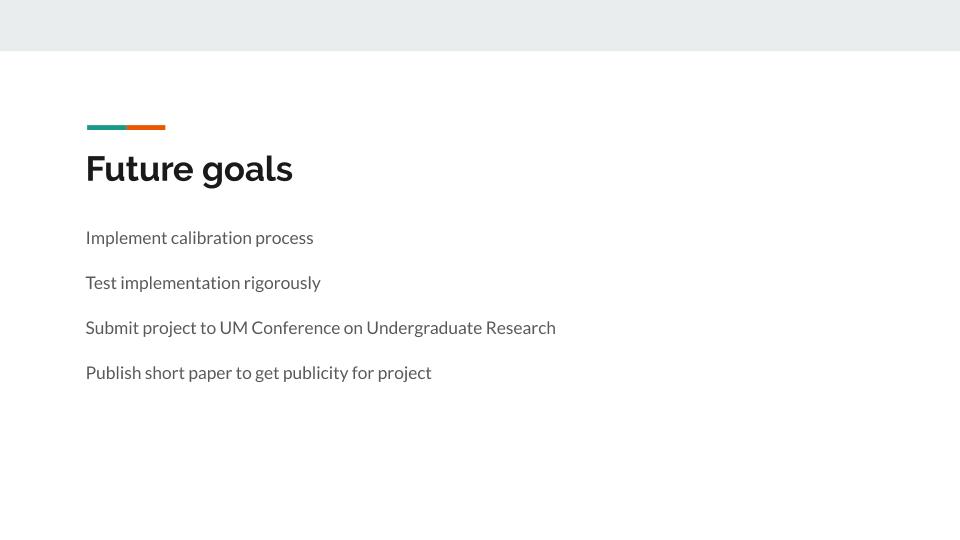
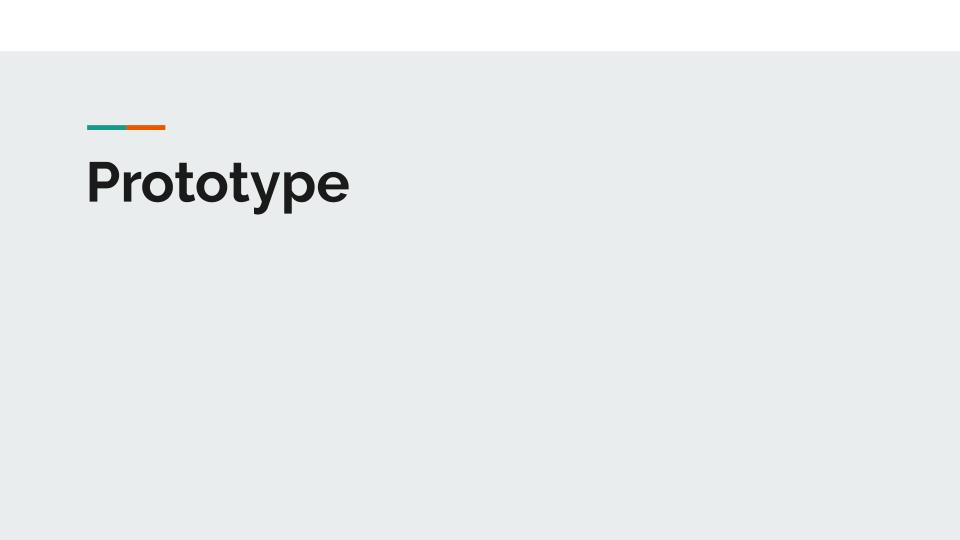
The team members hereby indicate by their signatures below that they have read and agree with the specifications of this document.



**Final Presentation**

****

****

****

**CSCI 426 Progress Report #: \_1\_**

**Group Name: \_\_\_The\_Skyentists\_\_\_\_**

**Dates that this progress report covers:**

August 26, 2019 (Beginning of Class) – September 23, 2019 (Present)

**What did the team accomplish during this time period? (be specific)**

The Skyentists were founded on August 28, 2019. Since the team’s inception, they created a team charter that defines the way we will work as a team as well as a method to resolve interpersonal conflicts. A project charter was created which gives a high level definition of the project. This includes what problem is being solved, background information required to understand the problem, and goals the software should accomplish. The team also had their first client meeting in which they established a good relationship with the client. Most of this meeting discussed the thoroughly written requirements document the client provided. The team has also gotten to know each other and is learning how to operate as a cohesive unit.

**What did the team plan to accomplish but fail to, and why?**

There have been minimal failures to date for the team. Although all assignments have been completed, they’ve been worked on mostly right before the deadline. As a result they were a bit rushed, which resulted in odd formatting and awkward phrasing.

**What particular challenges did the team face?**

The team faced a few scheduling challenges when attempting to have a team meeting outside of class time. We also had difficulties understanding the client’s complex ecological and biological workflow since the team does not have much experience in these fields.

**What accomplishments is the team most proud of?**

The team name of The Skyentists. The team is also proud of the project that we volunteered to work on due to the potential to expand our knowledge, as it will be challenging but rewarding once the software is written.

**What will the team accomplish during the next reporting period?**

The team will accomplish certain objectives by the next reporting period. These objectives include the requirements gathering documents, meeting with the client again, and beginning to design the software itself. We will continue to grow our knowledge and develop team chemistry.

**Any other concerns the team would like to mention?**

None Currently

**CSCI 426 Progress Report #: 2**

**Group Name: The Skyentists**

**Dates that this progress report covers:**

9/23/19 - 10/14/19

**What did the team accomplish during this time period? (be specific)**

The team completed the requirements gathering document and began rough drafts of systems modeling. We also met with Arthur to verify that our requirements gathering document met his standards.

**What did the team plan to accomplish but fail to, and why?**

Sometimes we leaned on Arthur’s work too much to make up for our deficits in understanding the project. Our communication could use some improvement as sometimes it was unclear the exact partition of responsibilities. We could also work ahead more than we do currently.

**What particular challenges did the team face?**

We had to read the provided requirements document in depth and understand it to complete both our requirements gathering document and systems modeling document. Coming up with an appropriate GUI is a challenge as Arthur has no preferences on the matter. It can be difficult to make a concise diagram that conveys the needed information.

**What accomplishments is the team most proud of?**

The systems modeling document has been satisfying to complete. We’ve transitioned from having some vague understanding of the calibration process to having a concrete workflow that gives us a deeper understanding of the software.

**What will the team accomplish during the next reporting period?**

We will finish systems modeling, low level prototyping, and an implementation plan. We will meet with the client again to review our documents so far, and begin work on the medium to high fidelity prototype

**Any other concerns the team would like to mention?**

None currently

**CSCI 426 Progress Report #: 3**

**Group Name: The Skyentists**

**Dates that this progress report covers:**

10/14/19 - 11/15/19

**What did the team accomplish during this time period? (be specific)**

We finished our low fidelity prototype, presented it to the class and received user feedback which we then classified by section of the project, and author. We created an implementation plan to maximize our ability to start coding on the first day of the spring semester. Significant progress has been made on our medium to high fidelity prototype, where we are making it as object oriented and clean as possible so it can be used for the actual project.

**What did the team plan to accomplish but fail to, and why?**

The team failed to explain what feedback will affect our design. This was an oversight as to how the team should classify feedback. We also failed to show Yolanda the specific changes we made to our documents (We wrote on the bottom of the document what changes we made, but that doesn’t allow for an easy comparison on Yolanda’s behalf.)

**What particular challenges did the team face?**

Explaining the process to lay folk in our presentation was difficult.

**What accomplishments is the team most proud of?**

Starting the medium to high fidelity prototype felt great. We’ve made all the necessary preparations to simplify the coding process and, as a result, development has been going smoothly.

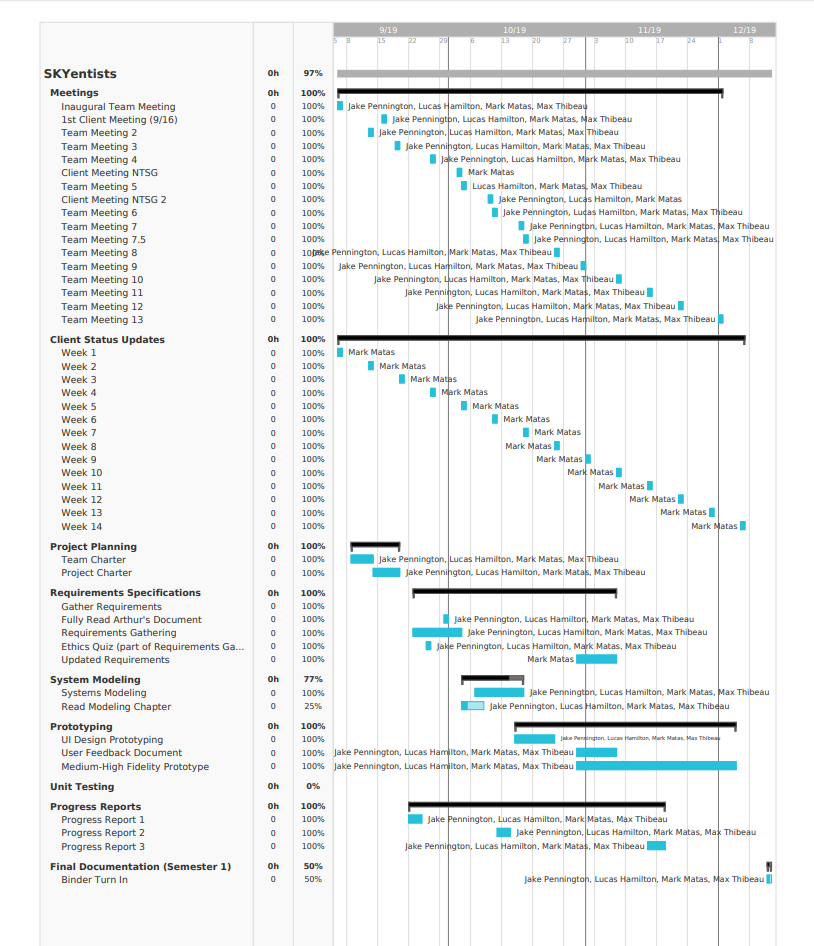
**What will the team accomplish during the next reporting period?**

The team will finish the medium to high fidelity prototype, make and give our final presentation for the semester, update all documents and assemble them in our final binder.

**Any other concerns the team would like to mention?**

None currently.

**Team Gantt**

****