UI Testing

Date: *April 17, 2020*

Prepared by: The Skyentists (Mark, Lucas, Jake, Max)

# 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 to use the SMAP L4C data, they would have to perform all the tasks other NTSG employees would perform but with less technical expertise.

# 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) via research grants. Currently the client, an employee for this company, oversees calibrating NTSG’s climate change model and edits or optimizes the data within it. If the client is unable to perform this calibration, then other or future NTSG employees will utilize this optimization software.

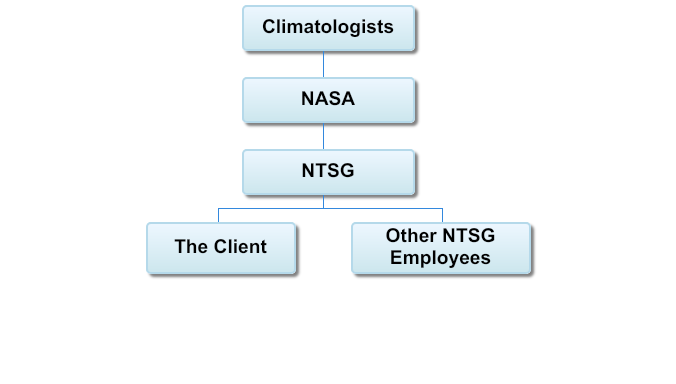


Figure 1. Hierarchy Chart

Figure 1. Hierarchy Chart

# User Groups

Table 1 below 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.

Table 1:

|  |  |  |
| --- | --- | --- |
| **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. |

# Test Plan

1. Test Purpose:

The usability testing is needed to ensure that the users of the project are able to complete the necessary tasks and not experience any errors during the workflow.

1. Problem Statement/Test Objectives:
2. User Profiles:

Pre-Virus Situation

Since our user(s) of the final product will not be a large number, all different types of people will be chosen. The team will attempt to get users that have a better knowledge of biological science or computer science as to better represent the small subset of final users. One group of users may be a University of Montana graduate computer modeling system class. The client (final user) and Professor Reimer will be testing. The team is aiming for 10 testing sessions that totals about 12 users.

Post-Virus Situation

Users will now be chosen by convenience, as the tests will have to be done remotely and/or in person with roommates/friends and one team member. The client and Professor Reimer will also be testing the interface of the project. The team is aiming for 6 testing sessions that totals about 6 users.

1. Test Environment/Equipment:

Pre-Virus Situation

The tests will be conducted in a public space (preferably the University Center on campus) to try and eliminate potential nervousness from the tester as it would be the tester and all four team members. The testing and recording of tests will be done on Max’s computer that uses a Linux operating system.

Post-Virus Situation

The tests will now be done remotely. The testers will be in a safe/comfortable place with at most one team member present and another member remotely present. The testing will be done on the present team member’s laptop which uses either a Linux operating system or the Ubuntu shell on a Windows operating system.

1. Roles:

Pre-Virus Situation

The roles defined below will mostly be done by the person, although there will be some movement if team members want experience with other roles.

Test Administrator (reading the test script, conversing with testers) will be done by Mark.

Technical Expert and Note Taker will be done by Jake.

Video Recorder, Timer, and Observer will be done by Max.

Data Logger/Note Taker will be done by Lucas.

Post-Virus Situation

Due to the COVID-19 virus shelter in place and social distancing, the roles for the team had to change. The test administrator role will be performed by all team members, as each member will have to remotely have a user test or be the only other person in the room with the tester. Data loggers/note-takers could be present in the room remotely via Zoom/Skype.

1. Evaluation Measures:

The testers will be evaluated with both performance and preference measures. Performance measures include the number of errors on tasks in the scenario list and a count of help access/questions. We will not be tracking the percentage of complete tasks or time to completion, as users will be able to finish all the tasks and the flow can iterate in many places. Preference measures include the ease of use and comments on the workflow. This type of measure will be shown in the post-test questionnaire and interview questions.

# Test Script

Hello and welcome! We are The Skyentists (Jake, Lucas, Max, and myself – Mark), a software development team that created the calibration software which is a process for an earth systems model used by NTSG on campus. This software is essentially an updated and better-documented code than what is currently used there. We’d like to take this opportunity to thank you for volunteering your time to help us test this new and exciting adventure into the world of plant carbon response. I will be reading from this script to ensure consistency between all of our participants.

Now a little about the calibration process. This software allows the user to filter carbon flux tower data from around the globe and optimize the parameters for a terrestrial carbon balance model that focuses on different plant types. Please note that our software is still in its development stage and that it may break down under certain circumstances. If that does happen, we will restart the program.

Understand that this exercise is to test the product and its usability and in no way implies your abilities. If at any time you feel uncomfortable, please inform us and we will terminate the exercise immediately. (We will be recording this exercise with a screen recorder on the computer so that we can gather as much information as possible from this session.)

Your opinion and ideas are important to us. Whenever possible, please speak your thoughts freely. Do not be concerned about offending us. If you forget to think aloud, I’ll remind you to keep talking.

As you’re working through this software, I won’t be able to provide help or answer questions on the usability of the program. We will provide some background information on things such as the plant types if necessary. This is because we want to create the most realistic situation possible. Even though we won’t be able to answer your usability questions during the exercise, please ask them. We’ll note your questions and answer them at the end of the exercise.

Do you have any questions currently?

# Checklist

Pre-Study:

* Laptop the test is run on has Ubuntu 18.04 LTS
* Ensure that setup.sh is ran to get all data files and dependencies

Study:

* Consent Form/Verbal Consent to record and take notes (will remain anonymous)
* Read Test Script to tester
* Take Notes

Post-Study:

* Make notes and appendix entry in UI Testing Document

# Data capture forms

All Data Capture forms are below:

|  |  |  |
| --- | --- | --- |
| **Participant #1 (Pilot Test)** | **Date:** 03/25/20 | **Time:** 10:30 – 11 AM |
| **Scenario #/ Task #** | **Team Member 1 Comment** | **Team Member 2 Comment** |
| 1 / 1 | * Begin button too small * Need label describing page * Confirmation that file has been chosen | * Begin button needs to be larger |
| 1 / 2 | * No need to return to home page * Previous page name should be the back button label | * Poor layout, buttons need to be better labeled |
| 1 / 3 | * Label closer to pull down menu * more clarity on default | * Do not know what path is for * No feedback for file selection * No back or home button |
| 1 / 4 | * Smaller box, bigger label |  |
| 1 / 5 | * Ask for feedback on graph about current params | * Grey out previous non-start |
| 2 / 6 | * No clarity on buttons * Unnecessary looping * New window open and ok to close(?) |  |
| 2 / 7 | * Bigger heading on current step * No clarity on how to view info |  |
| Misc | * Less verbose task list * Ass scenarios to task list * Add more context | * UI should hold hand * Add number in process * Avoid “navigate to..” in task list * Clear heading and steps throughout process |

----

|  |  |  |
| --- | --- | --- |
| **Participant #2** | **Date:** 04/09/20 | **Time:** 12 – 12:30 PM |
| **Scenario #/ Task #** | **Team Member 1 Comment** | **Team Member 2 Comment** |
| 1 / 1 | * Easily navigates to valid config file | * Liked opening screen |
| 1 / 2 | * Add header for pft chosen | * Change from 0-7 indexing to 1-8 indexing |
| 1 / 3 |  |  |
| 1 / 4 | * Outlier graph needs better spacing * X labels should be number of days/window size, not GPP or RECO |  |
| 1 / 5 |  | * Window size typically 4-10 |
| 2 / 6 |  | * Some nan/high numbers still in data |
| 2 / 7 | * Found out able to exit on own |  |
| 2 / 8 |  |  |
| 2 / 9 |  | * Re-display should use optimized GPP and RECO, not original everytime |
| 3/ 10 | * Likes slider for hyperparams |  |
| 3 / 11 |  | * Better exponential curve for ramp |
| 3 / 12 | * No need for graph to report differences, can use table | * Can get away with just graph |
| Misc | * GUI = “good” * Likes step numbers in process * More font unity (larger and SansSerif) | * Likes flow of GUI * Varying font sizes * Certain buttons have too small of font |

----

|  |  |  |
| --- | --- | --- |
| **Participant #3** | **Date:** 4/05/20 | **Time:** 7 – 7:30 PM |
| **Scenario #/ Task #** | **Team Member Comment** |  |
| 1/3 | * Unsure of what is the opening screen |  |
| 1/4 | * Doesn’t click ‘proceed’ * Confuses plot API with buttons for moving on |  |
| 1/5 | * Thought this was already done in the previous screen. |  |
| 2/7 | * Had some trouble finding the button |  |
| 2/9 | * Confused redisplay button with back button. |  |
| 4/15 | * Not sure of difference between exit button and ‘X’. |  |
| Misc | * Was uncertain of ‘where’ she was in the program at times. |  |

----

|  |  |  |
| --- | --- | --- |
| **Participant #4** | **Date:** 4/05/20 | **Time:** 8 – 8:30 PM |
| **Scenario #/ Task #** | **Team Member 1 Comment** |  |
| 1/4 | * Failed to proceed |  |
| 2/7 | * Had some trouble finding the button. |  |
| 2/8 | * Tried to have no parameters selected, program properly handled the error. |  |
| 3/10 | * Liked the slider. |  |
| 3/11 | * Didn’t like new frame. |  |
| 4/13 | * Had invalid input, program handled it properly. |  |
| Misc | * The screen of this session was recorded, and can be found in the ‘testing’ folder. |  |

# Scenarios & Task lists

You are an earth systems data model engineer at the Numerical Terradynamic Simulation Group (NTSG) at the University of Montana. The goal is to optimize parameters for both GPP and RECO for the plant functional types to eventually be used by the model.

1. Begin the calibration process by selecting a valid configuration file of type “.cfg”
2. Select a PFT that is not the default
3. The wrong configuration file may have been selected, so return to the opening screen
4. Change the Window Size to 0.4 to smooth the GPP outlier data
5. Smooth RECO outliers by changing Window Size to 12.7

The Gross Primary Production (GPP) is the amount of carbon stored in a certain area in a certain time for a specific plant type. The GPP optimization process displays graphs of GPP against its eight specific ramp functions and allows the user to iteratively optimize each one.

1. Begin GPP optimization process by viewing all the GPP ramp functions (3)
2. Display the optional graph of GPP vs. Emult and exit this page
3. Choose what GPP parameters to edit and display additional information on the GPP parameters
4. Re-display the GPP ramp functions and then return to choose what parameters to optimize

The Ecosystem Respiration (RECO) is the total amount of carbon dioxide that is being emitted by a specific plant type and soil microbes. The RECO optimization process displays graphs of RECO against its four specific ramp functions and allows the user to iteratively optimize each one, similar to the GPP optimization process.

1. Begin the RECO optimization process by entering or sliding values of 0.75 for Prh and 0.5 for Pk
2. View all the RECO ramp functions (2)
3. Display differences in the optimized RECO parameters

Soil Organic Carbon (SOC) is also an important calculation process as the model depends on the amount of carbon assimilated by the plants and emitted by SOC decay.

1. Display the graph of Estimated SOC vs. Calculated SOC
2. Specify the number of Numerical Spin-Up iterations
3. Select another PFT to go through the process again
4. GPP and RECO optimization has been completed, exit the program

# Post-test questionnaire

The answers, provided by the testers, to the post-test questionnaire are included in the Appendix section (Section 15).

1. On a scale of 1-5 with 1 (easy to follow) and 5 (hard to follow), how would you rate the overall flow of the program?
2. If you could change one thing about the design of the project, what would it be and why?
3. How likely would you be to view another program that has a similar design?
4. On a scale of 1 (worst) to 5 (best), how do you like the layout of the pages?
5. What are the major issues which will cause future users to be frustrated, if any?
6. On a scale of 1 (worst) to 5 (best), how do you think the user interface design accurately reflects the purpose of the software?
7. On a scale of 1 (not repetitive) to 5 (very repetitive), how repetitive does the design feel?

# Post-test interview questions

The answers, provided by the testers, to the post-test questionnaire are included in the Appendix section (Section 15).

1. Overall, what was the best thing about the usability of the program?
2. Overall, what was the worst thing about the usability of the program?
3. How would you improve XX page? (where XX is problem pages)
4. What aspects of the design would you improve?
5. Did you find yourself having to click “too much”?
6. If you were a scientist, would the design and flow be easier to understand/work with?
7. Any other feedback/notes?

# Conduct Testing Sessions, being mindful of the following requirements..

1. Using your test plan and the testing materials you prepared, conduct usability testing with 10 or more end users. You should have 6 or more individual end user testing sessions and 2 or more sessions with a pair (2) of users. Pair testing will give you first-hand experience with the differences (advantages, disadvantages, etc.) of testing with 1 user vs. 2.
2. **All** team members must be present for **all** testing sessions.
3. The tasks you give your users should be written down and not presented orally.
4. Capture each testing session using screen capture software, including audio.
5. You do not need to use a formal consent form for this study. However, you must be sure to fully explain to your users what the test is all about, why you are conducting it, what you will do with the information you gather, how you will handle the recordings, etc.
6. The recordings you take are to be used by team members for analysis only, and possibly to show snippets of to the class. They **MUST** all be destroyed by you at the end of the semester. There is to be NO dissemination of video, data or results to the public.
7. Remember to reassure your users that the purpose of the test is so that you can evaluate your design and that *they* are not being evaluated in any way (if they have problems completing a task, it is a weakness in the design and not an indication of the user’s abilities).
8. Encourage your users to think aloud as they work through the tasks (if participants are working alone, you may have to continually prompt them to do this throughout the session).
9. Be silent during the testing unless your users get irretrievably stuck.
10. Remind your users that their participation is greatly appreciated and totally voluntary. They can stop the test at any time if they wish.
11. When the testing is complete (i.e., when the user is done working through the tasks you give them using your system), follow-up with your questionnaire and some open-ended interview questions (i.e., what did users like, what didn’t work so well, what suggestions for improvement would they offer, etc.).

# Testing Results

After analyzing the notes on the overall comments, each individual task from the task list is summarized below in Table 1.

Table 1:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scenario #** | **Task #** | **Task Description** | **Overall Comments** | **Future Designs** |
| 1 (BASIC) | 1 | Begin process by selecting configuration file | This generally worked well. | No change. |
|  | 2 | Select a PFT | Good. | No Change |
|  | 3 | Return to Opening Screen | Confusion about what the opening screen was. | Better labels for our buttons. |
|  | 4 | Changing Window Size to smooth GPP outliers | A little hard to find. | Larger font / input field. |
|  | 5 | Change Window Size again to smooth RECO | Unclear what window this is. | Better labels. |
| 2 (GPP) | 6 | View GPP Ramp Functions | Unclear what is part of the GUI and what is part of the plot API. | Better labels, trim the fat on the plot. |
|  | 7 | Display optional graph of GPP vs Emult | Unclear what is part of the GUI and what is part of the plot API. | Better labels, trim the fat on the plot. |
|  | 8 | Edit GPP parameters to optimize after viewing additional info | The underlined label was confusing to multiple users. | Remove underline. |
|  | 9 | Re-Display GPP Ramp Functions | Unclear what is part of the GUI and what is part of the plot API. | Better labels, trim the fat on the plot. |
| 3 (RECO) | 10 | Enter values for RECO hyperparameters (Prh and Pk) | Good. | No change |
|  | 11 | Display RECO Ramp Functions | Unclear what is part of the GUI and what is part of the plot API. | Better labels, trim the fat on the plot. |
|  | 12 | Report differences in optimized RECO params | Not sure what the graph was meant to show. | Use a table instead. |
| 4 (SOC) | 13 | Display SOC graph | Button was hard to find. | Make it bigger. |
|  | 14 | Specify Numerical Spin-Up iterations | Good | No change |
|  | 15 | Select another PFT | Good | No change |
|  | 16 | Optimization Completed, Exit Program | Users were not sure if they should use the exit button or ‘X’. | Add a tool-tip that explains the exit button / program the ‘X’ to do the same thing. |

# Reflection

Testing is an integral part of software development and user-interface testing is no exception. Prior to the global pandemic, we did not have many user tests performed, so doing them during the quarantine created several difficulties. Of these, the most glaring was the inability for the entire team to be present during the live testing session. One way we attempted to alleviate this was to record the screens of the testing sessions. While this does provide useful information, it doesn’t tell the full story. For example, a user’s frustration is often difficult to discern from the movement of the cursor and so are things the user likes. We also had one of our recordings get corrupted due to a software error, and we can’t just have the user do it again as they already know what to do at that point. Despite this, we all learned al lot about the testing aspect of the development process through conducting these tests in addition to learning about our software.

We learned that a lot of people have no experience with Matplotlib, including people who work with a lot of scientific software. This was surprising to us. We also discovered the importance of fonts. Before testing, no one considered using anything other than the default font for anything in the GUI, now we realize how important it can be in helping the users to understand the navigation of the software. Another thing we learned was the importance of button placement. A lot of users had trouble with our buttons. After one testing session, we requested a user to try to break the software. He was unsuccessful, and that was a very proud moment for our team.

The most useful feedback came from Arthur. Obviously, this is because he is the end user, has an intimate knowledge of the calibration process and has a ton of experience with similar software. He was also the only person we had test the command-line-interface as it is especially difficult to find people who have used a command-line (especially during this pandemic). He provided a lot of useful feedback and made sure we were going in the right direction.

This has been a trying time for all of us, but we were still able to get a lot out of the testing. We learned a lot about the process and our software while doing so. While it would have been nice to conduct these in person, the situation has given us a unique opportunity to get hands on experience with remote testing that we would not have otherwise had. Overall, it was a good experience.

# Execution and Acknowledgement

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| [Jake Pennington 4/17/20] |  | [Insert name & date here] |  | [Insert name & date here] |

A close up of a logo

Description automatically generated

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

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

# Appendix

Questionnaires:

Participant 1:

1. On a scale of 1-5 with 1 (easy to follow) and 5 (hard to follow), how would you rate the overall flow of the program? 3
2. If you could change one thing about the design of the project, what would it be and why? Larger, clearer text to help navigate through the program.
3. How likely would you be to view another program that has a similar design? Unlikely
4. On a scale of 1 (worst) to 5 (best), how do you like the layout of the pages? 3
5. What are the major issues which will cause future users to be frustrated, if any? The navigation was very hard to follow.
6. On a scale of 1 (worst) to 5 (best), how do you think the user interface design accurately reflects the purpose of the software? 3
7. On a scale of 1 (not repetitive) to 5 (very repetitive), how repetitive does the design feel? 4

Participant 2:

1. On a scale of 1-5 with 1 (easy to follow) and 5 (hard to follow), how would you rate the overall flow of the program? 2
2. If you could change one thing about the design of the project, what would it be and why? Warning on the ‘X’ button, accidentally closed the program and had to restart.
3. How likely would you be to view another program that has a similar design? Likely
4. On a scale of 1 (worst) to 5 (best), how do you like the layout of the pages? 5
5. What are the major issues which will cause future users to be frustrated, if any? Premature exit has no warning.
6. On a scale of 1 (worst) to 5 (best), how do you think the user interface design accurately reflects the purpose of the software? 5
7. On a scale of 1 (not repetitive) to 5 (very repetitive), how repetitive does the design feel? 4

Participant 3:

1. On a scale of 1-5 with 1 (easy to follow) and 5 (hard to follow), how would you rate the overall flow of the program? 4
2. If you could change one thing about the design of the project, what would it be and why? Better labels to make navigation easier.
3. How likely would you be to view another program that has a similar design? Somewhat Likely
4. On a scale of 1 (worst) to 5 (best), how do you like the layout of the pages? 3
5. What are the major issues which will cause future users to be frustrated, if any? The plot buttons are confusing.
6. On a scale of 1 (worst) to 5 (best), how do you think the user interface design accurately reflects the purpose of the software? 4
7. On a scale of 1 (not repetitive) to 5 (very repetitive), how repetitive does the design feel? 5

Participant 4:

1. On a scale of 1-5 with 1 (easy to follow) and 5 (hard to follow), how would you rate the overall flow of the program? 3
2. If you could change one thing about the design of the project, what would it be and why? No popups, was unsure if closing them would close the program.
3. How likely would you be to view another program that has a similar design? Likely
4. On a scale of 1 (worst) to 5 (best), how do you like the layout of the pages? 4
5. What are the major issues which will cause future users to be frustrated, if any? None
6. On a scale of 1 (worst) to 5 (best), how do you think the user interface design accurately reflects the purpose of the software? 3
7. On a scale of 1 (not repetitive) to 5 (very repetitive), how repetitive does the design feel? 5

Data:

Users typically took 20-25 minutes to complete the tasks.

Users got confused with the plot buttons often.

Some users had difficulty in navigation.

The flow received the worst ratings.

Recordings:

All recordings done can be found in the ‘testing’ folder.