**Software for the Teaching of Organic Chemistry Compounds and their Nomenclatures**

Chris Lansing

Tim Elam

Roland Heintze

John Gibbons

**(need table of contents and cover page)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision Number:** | **Revision Date:** | **Author:** | **Summary of Changes:** |
| 1 | 2-28-13 | John Gibbons | Added revision page and reworked/reworded first half of document. |
| 2 | 2-29-13 | John Gibbons | reworked/reworded second half of document and added domain analysis. |
| 3 | 3-26-13 | John Gibbons | Third revision. |
| 4 | 4-9-13 | John Gibbons | Fourth revision. |
| 5 | 4-25-13 | John Gibbons | Changed requirements and platform information. |
| 6 |  | Roland Heintze | Finish domain analysis and final revision. |

# Client:

Dr. Shainaz Landge, of the Georgia Southern University Chemistry Department. A professor of organic chemistry and principles of chemistry. Organic chemistry is a branch of chemistry that deals with the structure, properties, and reactions of compounds that contain carbon. Organic chemists can create new molecules never before proposed which, if carefully designed, may have important properties for the betterment of the human experience. Dr. Landge teaches students the current understand of organic chemistry and sets them on a path to further develop within this field. As a principles of chemistry professor as well, Dr. Landge explains and shows the constituents of a substance and their attributes as well as possibly benefits and detriments they have or may have if combined with other elements.

**The Task to be Undertaken:**

The project entails creating an animation that will demonstrate, through animations, the proper naming of organic compounds. The program will allow the user to arrange carbon elements and their bonds into a proper molecule. The user can also choose to randomly generate the organic compound with one button click. It will then display a short animation in accordance with Dr. Landge’s present teaching analogy of a mailman traveling a route in a specific manner which follows the algorithm chemists use for the proper identification and naming of organic compound molecules.

**Domain Analysis:**

In preparation for developing this software our group researched to see if any other software existed that would meet or partially meet the requirements given to us by Dr. Landge. .....(Roland finish please)

**Benefits:**

The benefit of this program is that it will give students an elaborate tool for learning the proper means of finding an organic molecule’s nomenclature. This will be used to reinforce the lecture they receive in class and make things more interesting and fun for the students. This will also allow the client to focus on other subjects in the classroom while students review the nomenclature conventions with this software at home. This has the potential to be a publicly released asset for other professors to use in the classroom or to be downloaded by people interested in chemistry and the naming conventions used. In summary, our software will offer the following benefits:

* + An automated teaching tool for students and professors.
  + An illustration of a simple analogy for a complex topic.
  + A means to quickly identify the name of a complex molecule for the public domain.

**Preliminary Requirements Analysis:**

1) Interface - An interface which allows the user to click in squares making up a grid to arrange carbon atoms and their bonds. Alternatively, they should be able to generate a random compound. It should then offer options to generate a proper nomenclature or to generate the animation.

2) Backend - A system must exist to discover the proper name for an organic carbon compound based on the user input or randomly generated compound. It must additionally provide a path for the purposes of animation.

3) Drawing - The software must be able to take the data from the backend and generate a proper animation.

**Suggested Deliverables:**

Management:

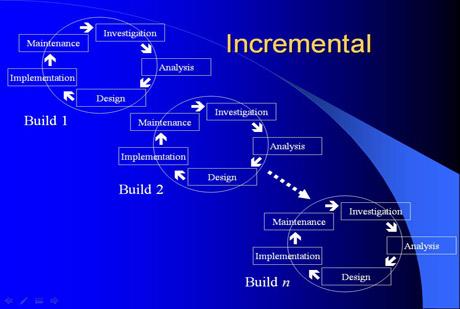
* + Requirements Analysis - A formal requirement. Will contain all requirements delivered from the client, minimum specifications the machines need to run the software, design constraints and user characteristics.
  + Status Reports - The customer will be kept updated with ongoing status reports that outline our progress with the program, mostly via e-mail. These reports will play a major role in keeping the customer involved and satisfied with the project.
  + Source - The project will be hosted on a private git repository at bitbucket. This site contains the source code as well as limits those who have access based on access levels granted by the creator of the repository. Any and all additions are time stamped as well as the user’s name who made the changes. Users can choose to have the newly added code highlighted. The creator then has the option to allow the code to remain permanent or revert the code back to the previous state.
  + User Manual - A guide or tutorial to explain the software and its use.

Technical.

* + A portable version of the software would need to be available.

**Process to be followed:**

The team will utilize the Incremental Model for the development life-cycle. In this way we can continually develop the software while having the client review parts of the implementation. In this way we can continually receive feedback to ensure that the client remains part of the development process. This will address any confusion or misunderstandings pertaining to the requirements or deliverables at any stage of development.



**Development Schedule:**

* **Milestone 1:** Feasibility Study (1/22/13): Begin Feasibility study draft. Begin Domain analysis research.
* **Milestone 2:** Feasibility Study (1/29/13): Final version of our groups Feasibility study.
* **Milestone 3:** Client Meeting (2/1/13): Begin extensive requirements gathering and design models.
* **Milestone 4:** Requirements Documentation (1/30/13): Begin constructing the requirements documentation and ensuring all requirements are fully documented.
* **Milestone 5:** Requirements Documentation (2/7/13): Final version of requirements documents with use cases each containing at least one use case scenario and GUI prototypes.
* **Milestone 6:** Project Schedule (2/8/13): Begin creating a comprehensive project schedule for development.
* **Milestone 7:** Project Schedule (2/12/13): Final version of project schedule with milestones.
* **Milestone 8:** Project Design (2/13/13): Begin creation of UML diagram and state diagrams. Start creating classes and variables with types.
* **Milestone 9:** Project Design (2/16/13): Final version of UML and State Diagrams are to be completed. All foreseen variable names and method names are to be included.
* **Milestone 10:** Software Quality Assurance (2/22/13): Begin constructing the checklists to ensure each stage of development is professional and are performed according to IEEE and standard Object Oriented Design principles.
* **Milestone 11:** Software Quality Assurance (2/28/13): Final versions of checklists with their own documents are to be completed.
* **Milestone 12:** Implementation (2/22/13): begin implementing the code in accordance to all documentation and designs.
* **Milestone 13:** Implementation (4/20/13): Desired date to have completed implementing all code to ensure plenty of time for testing.
* **Milestone 14:** Test Plan (4/1/13): Begin designing test plan and discerning possible areas of concern.
* **Milestone 15:** Test Plan (4/21/13): Final version of test plan.
* **Milestone 16:** Alpha Testing (4/22/13): Conduct alpha testing with the client to flesh out any functional or design issues the client may have with the current software.
* **Milestone 17:** Beta Testing (4/24/13): Conduct beta testing to ensure all previous issues have been resolved and the program functions according to the client’s wishes on platforms the client previously stated for the requirements documents.
* **Milestone 18:** Test Reports (4/27/13): Completion of all test report documentation.
* **Milestone 19:** Notebook (4/28/13): Completion of the notebook and all documents within to submit on schedule.
* **Milestone 20:** Presentation (5/7/13): Completion of powerpoint and rehearsal for presentation.

**Outline Plan:**

**Requirements Gathering:** We will need to communicate with the customer to draft the requirements document. This document will include everything that the customer believes we need in the program and we will follow it precisely. Our team may provide additional suggestions to the customer as to requirements and functionality of the program in order to deliver a more robust program. However, the requirements will ultimately be up to the customer. To avoid any future disagreements with the customer, we will have the client sign off on the requirements document confirming that the stated requirements will be what our team will be developing from.

**Design:** We will create the necessary diagrams for our program, such as UML, class, and use case diagrams. This way we will be able to flesh out the entire program and have a solid understanding of how we will development it. We will be able to quickly and efficiently complete development by following the models, and all members of the team will have an outline to follow should they ever misinterpret the code or its design.

**Customer Feedback:** Though we will be getting feedback from the customer throughout the entire development cycle, it is crucial that we get detailed feedback at this point from the customer. We want to ensure the customer and our development team are in full agreement and understanding on all aspects of the project before we start development in regards to how the program is going to function and the desired final deliverables.

**Development:** The overall structure and function of the program will be implemented from the design documents. The bulk of the development for this project will be creating the necessary animations to illustrate the story that the customer requires. Some or all of the code that Dr. Cook has created for the project may need to be rewritten. This process should not be time consuming due to the fact that all of the algorithms that are used, which we will be using to name the compounds, are all well defined, scientifically precise and straightforward.

**Testing:** Once our team is satisfied with the usability and functionality of our program, we will begin unit and integration testing. Unit testing will be done using PyUnit to verify that all of the classes, methods, objects, and variables are functioning correctly. Integration testing will be done by running many (possibly automated) simulations of our program to ensure that it does not fail on any edge cases. Changes to the program may be made as necessary during this phase in order to prepare it for release.

**Release:** Once our product is finished, we will deliver a complete, portable, copy and paste-able version of our software to the customer via usb drive. This program will be small enough to transfer to students’ computers via usb drive, a shared folder on the school network, or allows for the professors to send the program directly to students. After the file is on a student’s computer, the program will run simply by double clicking on the icon of the file from the desktop.

**Maintenance:** While currently there is no planned ongoing maintenance for the software after the class is over, the customer will have our team members’ emails in case future questions or concerns, attempts to get one or more members to continue development, or to fix any bugs in the software. The customer will also be given the entire source code for the project. This source code will be school property and the customer may enlist the help of any student(s) to take over the project once our team is finished working with it.

**Visibility Plan:**

**External:** It is vital that at least one member of our team make weekly/bi-weekly check-ins with the client to ensure the program is constructed to all specifications and requirements and that any new questions or concerns from the client are addressed. The base code that generates the molecules has already been written by Dr. Cook and has been approved by the client. Our development team will be constructing an animation to explain how each combination is named. With the procedures being a well defined processes, we have to ensure that the integrity of the naming process stays intact. Also, we need to ensure that our client’s models come to life in a way that will best convey the messages to the students properly who will be using this program as a study tool. One or more members of our group will report our progress to the client and explain current status of designs and coding up to that point. If at all possible, our team wishes to show working examples and demonstrations of functionality of the software within one and a half months into development. We will make sure the client agrees to and signs off on each step of the development process such that there will be records of approval and satisfaction.

**Internal:** In addition to regular contact with the client, regular contact between the group members is vital in completing our task on time, professionally and to ensure the clients’ wishes are fully met. The way our team has set up communication involves using IRC for chat purposes. Our group is also using google documents to keep track of all documentation and participation. Finally, the team is using EGit to keep track of the source code, as well as times and sections of code that were recently edited and by whom. This gives each member of our group instant access to all sections of the project from any machine that is connected to the internet. We will also schedule meeting times and places for face to face group work/discussions when necessary.

**Risk Analysis:**

**Time Risk:** As our team has limited time to complete this task, time management will be crucial. Our group will be setting up multiple deadlines to ensure work is completed early or on time. A checks and balance system will be implemented where group members can check on other group members’ progress to ensure deadlines will be met. Roland has the ability to reassign work or assign additional group members onto a section of work anytime he feels necessary to help ensure the deadlines are met.

**Existing Software Risk:** The existing software our group will be working from was created by Dr. Cook. If it takes longer than expected for our software development team to fully grasp the code and its functionality, Dr. Cook has offered to help explain those areas to one or more members of our group at any point he is available. However, Roland and Tim have spoken with Dr. Cook already and stated that they have a good grasp on Dr. Cook’s code and its functionality and thus our group does not foresee this being an issue.

**Resource Risks:** The main resource risk in this project would be if one or more members of our group decide to not manage their time properly or decide not to contribute to the project at all. If either or both of those events transpire then the whole project could fall behind or run the risk of failure to be completed within the given time schedule. With the above mentioned checks and balance system as well as the group leader, Roland, checking on progress at certain intervals, our group hopes to prevent this or become aware of this situation far enough in advance to compensate for it. This project consists of extensive animation, where our team will be relying on Roland and Tim, who have taken courses and have experience in this field, to do the majority of the coding. If both those members get sick and/or become unable to code, that situation would cause our team and the project as a whole serious issues. However, if one becomes unable to code, the other would be able to compensate for the work load with support from Chris and John. As far as the documentation is concerned, all four members of the group have experience with creating, editing and completing documentation, so our group should not encounter any issues pertaining to this area. Additional risks include the customers’ hardware and additional costs revealed after the initiation of the project. The hardware will not be an issue because the machines in the school labs as well as the majority of machines owned by the students and professors will be able to run the .exe the program will be made into. However, if the .exe file does not function properly, the Python interpreter Py2exe will be used to allow the program to run on those machines. On top of the interpreter, the machines will need to possess the processing power to generate and run the animations. As for additional costs, there are none. Infact, this program will be completely free to develop. All programs and/or applications needed to run this program are free to download and use. Also, the machines in the labs do not need to be upgraded for this program and have dedicated hours for students to use so there is no additional costs that may occur after development.

**Changing Requirements:** It is possible that during the semester, the client may want more animations, higher end graphics or a different sequence of animations. To help ensure that our group can deliver the best possible software to the client, the group leader will have the client sign off at the beginning of the development on the requirements our group will be given. If the client wishes to add additional requirements, our development team will assess the changes and provide a detailed explanation as to which new requirement(s) are possible to implement in the timeframe allotted and which are not. As for the client being sick, this may delay the progress reports done in person which entail showing current functionality but one member of our group can still email the current progress and possibly include screen shots of the animations depending on their completion. Another factor would be if the client becomes disinterested in the project at any period during development. If that occurs our team would work with closely with the client to hopefully re-inspire them back into the project by doing demonstrations (if possible) and by showing current models of what the program is currently doing and what it will be doing as well as going into how students can greatly benefit from the lessons this program can provide them.

**Probable Technical Requirements:**

The software will be required to operate on multiple platforms to facilitate its’ availability to a wide range of students from many different classrooms. In order to meet this requirement, our group has chosen to write the application in Python 3 with PyQt4. This way the program will be capable of running on all platforms when converted to a .exe file at compile or with a Python Interpreter called Py2exe. This includes virtually all available platforms to the user population. Our application requires no additional applications or libraries outside of its own bytecode so portability will be transparent on the development end.

**Conclusion:**

This is a great project with outstanding rewards. The costs for this project is non-existent. The project, within the specified guidelines and requirements, is completable within the given timeframe of this semester. When this project is finished, it will deliver a product to the customer that will satisfy the clients’ requirements and expectations for the software and more. Also, our group believes that it will be a program in which other professors and students will enjoy and benefit greatly from.