LoCuS
Laboratories for Computer Science

Lab Specification Authors’ Guide

Matthew Johnston, Rama Rao Cheepurupalli, and Richard Snodgrass

Department of Computer Science
The University of Arizona

last modified: July 8, 2010
Table of Contents

Overview

I. General Appearance and Functionality
   A. Launching the Application
   B. Splash Page
   C. Lab Interface
      1. Chalkboard
      2. Bookshelf
      3. Lab Notebook
      4. Lab Bench
      5. Data Cabinet
      6. Apparatus Shelf
   D. Parts of a Written Lab
      1. Introduction Section
      2. PreLab Section
      3. PreLab Quiz
      4. Lab Section
      5. Lab Quiz
      6. PostLab
      7. Open Lab Section
   E. Guidelines for Lab Versions by Grade Level

II. Writing a Lab in XML
   A. Getting Started: XML
      1. Elements
      2. Schemas
      3. Comments
      4. Attributes
   B. Basic XML Structure of a Lab

III. Lab Elements
   A. Lab
   B. Metadata
   C. Visibility
   D. Introduction Section
   E. PreLab Section
   F. Practice Question
   G. PreLab Quiz
   H. Lab Section
   I. Lab Quiz
   J. PostLab Section
K. Open Lab Section
L. Prose
M. Table
N. Image
O. Movie
P. Launch Apparatus
Q. Multiple Choice Question
R. Fill in the Blank
S. Free Response
T. Turn in Lab

IV. Generated Files
   A. Saved Lab
   B. Answer Sheet
   C. Grading Rubric

V. Grading
I. General Appearance and Functionality

The software contains a set of labs that investigate theories in computer science. The software runs in Java from a website. The basic user interface is the same for all labs and simulates an actual science laboratory environment. Different labs are written in XML to be executed within this framework. Each lab contains apparatuses that are used to perform specific functions (like display a graph) that can be manipulated by the user. The user records their findings and answers quiz questions in a lab notebook that can be saved and turned in. The software generates data files, completed labs, and answer sheets that are uploaded and saved to a website. Each lab has different versions that are appropriate for different grade levels (middle school through undergraduate college). All of these components will be explained and exemplified within this document.

This section describes the aspects of the lab software that are universal across all of the written labs.

A. Launching the Application

The user opens the software application by clicking an icon.

B. Splash Page

When the application launches, a splash page is given. The splash page contains:

• title of the software
• authors
• location
• NSF grant information?
• date created
• last modified
• an image of a door at the bottom right of the screen
• an "About This Software" button

Functionality:

It is shown for 5 seconds after which the image of the lab door is shown to open (with an animation) and the user sees the full Lab Room (Lab Room appearance is explained below).

When the "About This Software" button is clicked, a dialogue box is displayed showing the following information:

• credits for all contributors, including for each apparatus
• Department of Computer Science, University of Arizona
• grant information
C. Lab Room

The lab room is what the user sees immediately after the splash page. The lab room emulates an actual science lab. Key elements of the lab room are:

- Chalkboard
- Bookshelf, containing lab notebooks
- Lab Notebook
- Lab Bench
- Data Cabinet, containing data files
- Apparatus Shelf

1. Chalkboard

The chalkboard is essentially a dialogue box that looks like a chalkboard. It is used as a communication link between the software and the user. It shows text given by the software intended for the user to read. *where is it located? as an item within the room, or a dialogue box that is only visible, front and center, when a message is given? This text could be in the form of:

- messages (e.g. "You must answer all quiz questions before checking your answers.")
- hints (e.g. "In order to solve this problem, you count the number of nodes and divide by the number of closed triads")
- prompts (e.g. "First, open a lab by selecting it from the bookshelf.")

Functionality:

Text on the chalkboard should appear to be written in chalk!

2. Bookshelf

The bookshelf contains the available labs (that is, unfilled and filled lab notebooks) that appear as books on the shelf. The bookshelf could be located as a separate element somewhere with the lab interface, or integrated into the lab bench. The bookshelf has separate shelves, for:

- blank labs, that have not been filled in
- saved labs

Functionality:

The user opens either a new, blank lab or a saved lab by clicking on the correspondingly named book on the bookshelf.
3. Lab Notebook

The *lab notebook* contains the current written lab. It looks like an actual notebook with lined paper. The notebook contains:

- background information
- instructions for what to do
- questions and quizzes
- space for users to write answers, notes, etc.

**Functionality:**

The notebook is interactive. Not only does it contain text and space for students to write answers, but it also contains clickable icons, when appropriate, that can:

- launch an apparatus
- open a data file
- save a lab
- open help

4. Lab Bench

The *lab bench* looks like an actual bench that one would see in a science lab. It is a work surface that contains

- lab notebook
- any apparatuses that have been launched

**Functionality:**

The lab notebook and apparatuses on the lab bench are clickable. When the lab notebook is clicked, the notebook springs forward and takes up most of the viewable area and becomes active (user can interact with it). When the user clicks an apparatus on the lab bench, it maximizes the window for that apparatus. When the notebook or apparatus windows are minimized, they appear back on the lab bench.

5. Data Cabinet

The *data cabinet* is a drawer (or drawers) that can be pulled out. The data cabinet could be off by itself somewhere or integrated into the lab bench (located under the bench, for example).

The data cabinet contains:

- *data files* included with the software (i.e., example graphs)
- user folders containing user-generated data files (i.e. graphs, tables, etc)
Functionality:

When a user clicks on the data cabinet, the drawer pulls out showing all of the files that have been stored there. Files could be located within folders (that look like actual folders with labeled tabs). Clicking anywhere else closes the data cabinet.

6. Apparatus Shelf

An apparatus is a lab device that can be used to do a specific task. For example, the Social Network apparatus is used to open and create graphs representing social networks. This apparatus also gives some statistics about the graph and can display graphs in different formats. The apparatus shelf is a set of shelves that contain images of all of the apparatuses. (It could be off by itself or integrated within the lab bench).

Functionality:

All apparatuses are seen on the shelf. However, apparatuses that are not used for a particular lab are greyed out and not clickable. So each lab will only have a handful of apparatuses that are full color and clickable. When a clickable apparatus is clicked, the apparatus is launched.

Graphs, tables, and figures produced by an apparatus can be "dragged and dropped" into a notebook. The user clicks and holds on the image and drags it to any blank area in the notebook. When the mouse button is released, the image pastes into the notebook, moving subsequent material down to make space for the image. (we need to discuss what size these pasted images will be…iconified or based on size within the apparatus? Can be resized?)

D. Parts of a Written Lab

The written lab is what appears in the lab notebook. Although there is some flexibility in how each individual lab can be written, there are certain aspects that are consistent across all labs, as described below.

1. Introduction Section

The introduction gives a short paragraph summarizing what the lab is investigating.

2. PreLab Section

Each lab has a prelab section. This section must be completed before they can begin the actual lab portion. For easy material, the prelab could be short, designed to be completed
in a few minutes. For more difficult material, the prelab could be more in depth, designed to be completed in one 50 minute class session. The prelab contains:

- background information
- practical applications of the theories presented
- practice questions and exercises where appropriate

3. PreLab Quiz

Each lab has a multiple choice quiz at the end of the prelab section. Students must pass this quiz in order to move on to the lab section. This ensures that the students have enough mastery over the material in order to perform the lab efficiently. The quiz can be retaken multiple times.

4. Lab Section

The lab section is where the student uses the scientific method in order to investigate a theory. Each lab must require the student to:

- form or identify a hypothesis
- test the hypothesis by experimental manipulation of variables
- record the results of their experiment
- analyze the results of the their experiment
- form a conclusion based on the results of their experiment (did the data support the hypothesis?)

The lab can also contain questions for the students to answer (multiple choice, fill in the blank, free response).

5. Lab Quiz

Each lab has a multiple choice quiz at the end of the lab in order to assess student learning. Quizzes should be about 10 questions long. Quizzes will be graded by the software.

6. Post Lab (optional)

If desired, a lab can have a post lab section containing additional questions for fun or for homework. These can be take-home questions that do not require the lab software in order to be answered. This is a good place for free-response questions that require the student to take some time and thought in writing.

7. Open Lab (optional)
If desired, a lab can include an additional section at the end titled Open Lab. Here, students are encouraged to explore the program in a more free-from way, with very little guidance. This section could be used in order to:

• encourage students to explore, tinker, manipulate the various apparatuses
• give the students a hypothesis to test that is different from the guided lab, and have them write up a lab in a less guided fashion
E. Guidelines for Lab Versions by Grade Level

Most of the labs will require three versions: “Beginner”, “Intermediate”, and “Advanced.” The three levels are based on the level of science class the different grades will be in. A good strategy is to work on the “Advanced” version first, then see how it can be simplified for the other levels. When possible, even simpler version for 7-8th graders will be written, but this may only be possible for certain labs.

The following table gives some guidelines for writing labs for different student levels.

**AP:** refers to high school students taking an Advanced Placement science course (usually 10th, 11th, or 12 graders)

**undergrad:** refers to students in a four year college or university taking an introductory science course

**introCC:** refers to students in high school or community college taking introductory community college credits

**upperCC:** refers to students in high school or community college taking higher level community college credits

<table>
<thead>
<tr>
<th></th>
<th>Beginner: 9-10 grades</th>
<th>Intermediate: 11-12 grades/ introCC</th>
<th>Advanced: AP/undergrad/ upperCC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>vocab</strong></td>
<td>• as little technical vocab as possible</td>
<td>• intermediate</td>
<td>• Unleash the jargon!</td>
</tr>
<tr>
<td></td>
<td>• remove or reword any jargon that isn’t mission critical</td>
<td>• 8 technical terms</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 3-5 technical terms</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>amount of written instruction</strong></td>
<td>As little as possible.</td>
<td>• intermediate</td>
<td>• written instructions can be more in-depth. • Background info can be more extensive.</td>
</tr>
<tr>
<td></td>
<td>• No big paragraphs (they won’t read).</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Background info shortened to the bare minimum.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• must be task-driven, not text- driven</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>visuals</strong></td>
<td>• use images/graphics wherever possible</td>
<td>• use images/ graphics wherever possible</td>
<td>• use images/graphics wherever possible</td>
</tr>
<tr>
<td><strong>timing</strong></td>
<td>• need 50% longer than 11th grader through undergrad</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>math required</strong></td>
<td>• basic algebra</td>
<td>• higher order graphing</td>
<td>• higher order equations,</td>
</tr>
<tr>
<td></td>
<td>• mapping</td>
<td>• matrices</td>
<td>• Solving equations using matrices</td>
</tr>
<tr>
<td></td>
<td>• geometry</td>
<td>• functions</td>
<td>• graphs, trigo, pre calculus</td>
</tr>
<tr>
<td></td>
<td>• linear equations.</td>
<td>• polynomial functions</td>
<td></td>
</tr>
<tr>
<td><strong>lab responses</strong></td>
<td>• no free response</td>
<td>• some free response</td>
<td>• many free response</td>
</tr>
<tr>
<td><strong>quiz questions</strong></td>
<td>• easy</td>
<td>• intermediate</td>
<td>• some difficult</td>
</tr>
</tbody>
</table>
II. Writing a Lab in XML

All labs are written following the guidelines explained above (Parts of a Written Lab, and Guidelines for Lab Versions by Grade Level. All labs will be written in XML (Extensible Markup Language).

A. Getting Started: XML

XML's design goals emphasize simplicity, generality, and usability over the internet. What follows is a very basic introduction to XML as we use it. For more information visit:

http://www.w3schools.com/xml/default.asp

1. XML elements

XML documents are composed of XML elements. In an XML document, elements are indicated by start tags and end tags. For an element, the start and end tags are the same, except the end tag has a "/". Start tags must always be closed by an end tag. Within the start and end tags is content.

Example:

Let's say we want to include a greeting:

Hello, how are you?

The XML for this would be:

<greeting>Hello, how are you?</greeting>

In this example, the element is greeting. <greeting> is the start tag, </greeting> is the end tag.

Tags are invisible on web pages, so for the above example, one would simply see this on a web page as:

Hello, how are you?

Elements are often nested within other elements creating an organized hierarchy. When writing XML, it is crucial that elements are properly nested.
Incorrect example:

\[<b><i>This text is bold and italic.</i></b></i>

This example shows improper nesting of elements. Since the start tag \(<b>\) was first, the end tag \(</b>\) should be last. The tags \(<i>\) and \(</i>\) should be nested within.

In XML, white space on the page is ignored. In order to start a new line, or to leave spaces, line breaks must be inserted using a \(<br/>\) tag and the end of the text. Otherwise, material will be shown on the page continuously, with no line breaks or paragraphs.

In XML, the user specifies their own tags; they are not predetermined. Elements are given tag names that are as descriptive as possible without being too long. XML tags are case sensitive. For our labs, we use a naming convention where all letters are lowercase EXCEPT the first letter of subsequent words making up the element name. There are no spaces in the names of elements.

Example:

\(<\text{labQuizQuestion}>\)

Only the first letter of words in the middle of the name are capitalized and there are no spaces between words.

2. XML schemas

Since element names are user-defined and simply contain text, an XML schema must be created that describes the structure of an XML document. An XML schema:

- defines elements that can appear in a document
- defines attributes that can appear in a document
- defines which elements are child elements
- defines the order of child elements
- defines the number of child elements
- defines whether an element is empty or can include text
- defines data types for elements and attributes
- defines default and fixed values for elements and attributes

There is a schema already in place for this software. When writing a lab, it is important to only use elements that are part of the defined schema. These elements and attributes are listed in Section III of this guide.
3. Including comments in XML

Sometimes, a writer might want to include comments within an XML document explaining what a particular element appears or what it does. In order to include useful comments, but have them not be part of the XML code, use this format:

Example:

<!--this content contains useful information but is ignored by the software-->

the tag “<!--” indicates that a comment will follow and the tag “-->” marks the end of the comment.

4. Attributes

XML elements can have attributes that contain additional information about that element. Attributes often contain information that is valuable to the software manipulating the element. Attributes are linked to an element by an = sign and the attribute itself is in quotes.

Example:

<step number="3">Now do this</step>

In the above example, the element <step> has an attribute, called number, with a value of “3”. This attribute sets it apart from other potential step numbers.

B. Basic XML Structure of a Lab

The parts of a written lab, as described above, are delineated by certain XML elements. These elements are containers that hold other elements. These elements can also affect the functionality of certain elements contained within. For example, multiple choice questions can behave differently depending on which section of the lab they appear. The following example shows the general structure of all the written labs. Lab writers should follow this general structure when creating labs. A detailed description of the elements within a lab is given in section III of this document.

Example:

<lab>
III. Lab Elements

Lab elements are XML elements that serve a number of functions, such as:

- delineating the different parts of the lab
- serve as containers for other elements
- allow certain content to be displayed in the lab Notebook

This software contains the following lab elements:

A. Lab <lab>

The <lab> element is the root element. The root element is the element that contains all other elements. XML requires all code to be contained within a single root element. So each lab begins with <lab> and ends with </lab>.

B. Metadata <metaData>

At the beginning of the lab XML is metadata giving the title of the lab, authors of that lab, when the lab was created, when it was last modified, and a brief description of the lab.

Example:

```xml
<metaData>
  <labTitle>Social Networks</labTitle>
  <author>Matthew Johnston</author>
  <author>Rama Rao Cheepurupalli</author>
  <author>Richard Snodgrass</author>
  <createdDate>June 10, 2010</createdDate>
  <latestModification>June 23, 2010</latestModification>
  <description>This lab explores aspects of social networks</description>
</metaData>
```

Functionality:

The metadata is not part of the actual written lab and is not visible.

C. Visibility <visibility>

The <visibility> element controls which aspects of a lab are visible to the user for a given lab. For example, not all apparatuses will be needed for all labs. Apparatuses that
are not used for a particular lab will be visible, but greyed out. In other cases, we may want to hide certain things altogether if they are not applicable to a lab.

Example:

```xml
<visibility>
  <dataFiles>
    <dataFile>
      <path>socialnetwork/Rex.sn</path>
      <title>People Rex Knows</title>
    </dataFile>
  </dataFiles>
  <labBench>
    <apparatus maxOpen="1">Social Network</apparatus>
  </labBench>
</visibility>
```

Functionality:

By default, data files and apparatuses will be greyed out or not shown unless otherwise specified by the Visibility lab element. Only the file `Rex.sn` with title "People Rex Knows" is shown. All apparatus are greyed out except the apparatus called Social Network (we still want the user to see the other apparatuses, even if greyed out, as a teaser for what other labs have in store for them). The attribute `maxOpen="1"` indicates that only one instance of the Social Network apparatus may be opened at one time. This prevents students from filling up the screen with multiple windows.

D. Introduction Section `<introductionSection>`

The `<introductionSection>` element contains the Introduction section.

E. PreLab Section `<preLabSection>`

The `<preLabSection>` element contains the PreLab section. The PreLab section contains, amongst others, two unique lab elements found only in the PreLab section: `<practiceQuestion>` and `<preLabQuiz>`. These elements affect the functionality of the multiple choice questions contained within them (described below under Multiple Choice Question).

F. Practice Questions `<practiceQuestion>`
The `<practiceQuestion>` element gives a multiple choice question, accepts a response, and gives feedback. This type of question is found only in the PreLab section. Practice questions have the following aspects:

- They are multiple choice questions interspersed throughout the text.
- They ensure that students are reading and understanding the material.
- They do not count toward students’ lab grade.
- When answered incorrectly, the software gives the correct answer and an explanation `<explanation>` for why it is the correct answer.

• `optional`: Depending on the lab, we will want an option to have a hint button that will return a clue as to how to answer the question. For Practice Questions, students can click this button at any time (before or after attempting to answer the question).

For an example of a practice question, and a description of its functionality, see the Multiple Choice Question section.

G. PreLab Quiz `<prelabQuiz>`

The `<prelabQuiz>` element gives multiple choice quiz questions at the end of the prelab. Students must pass this quiz in order to go forward with the lab. Passing a quiz entails getting a certain number of questions correct. The quiz acts as a checkpoint. The quiz can be taken multiple times. Hints, if included, are only available when the student is retaking a quiz, they are not given the first time.

For an example of a PreLab Quiz question, and description of its functionality, see the Multiple Choice Question section.

H. Lab Section `<labSection>`

The `<labSection>` element contains the Lab section. The Lab section contains, amongst others, the unique lab element `<labQuiz>` which affects the functionality of the multiple choice questions contained within it (described below under Multiple Choice Question).

I. Lab Quiz `<labQuiz>`

The `<labQuiz>` element contains a lab quiz. Each lab will have a lab quiz consisting of several multiple choice questions. These questions are graded by the software.

For an example of a Lab Quiz question, and an explanation of its functionality, see the Multiple Choice Question section.
J. PostLab Section (optional) <postLabSection>

The <postLabSection> element contains the PostLab section, if written.

K. Open Lab Section (optional) <openLab>

The <openLabSection> element contains the Open Lab section, if written.

L. Prose <prose>

The <prose> element adds text in the Notebook. This includes all background information, headings and subheadings, etc. Note that not all words seen on a lab Notebook are prose. For example, the text that makes up questions is a separate element within the lab question element.

Example:

Let's say you wanted the following text to be shown in the Notebook:

Social networking analysis has been used in a number of different fields.

In XML, this would be:

<prose>Social networking analysis has been used in a number of different fields.</prose>

The author can specify certain aspects of how the text will appear, i.e., italics, bold, color, and font, as shown in the following examples:

1. Italic <i>
   This tag gives italic text

   Example:  This text is italic.

   <prose><i>This text is italic.</i></prose>

2. Bold <b>
   This tag gives bold text

   example:  This text is bold.
<prose><b>This text is bold.</b></prose>

3. **Underline** `<u>`
This tag gives underlined text.

**Example:** This text is underlined.

<prose><u>This text is underlined.</u></prose>

4. **Subscript** `<sub>`
This tag gives text in a subscript.

**Example:** This text has a subscript in it.

<prose>This text has a <sub>subscript</sub> in it.</prose>

5. **Superscript** `<sup>`
This tag gives text in a superscript.

**Example:** This text has a superscript in it.

<prose>This text has a <sup>superscript</sup> in it.</prose>

6. **Font Type, Size, Color**
Font type, color, and size are changed by assigning attributes using `style=`. This is unlike previous examples where the style of the text was changed using element tags.

**Example:**

My name is Jim, I am a teacher in Tucson.

<prose style="font-family:arial;color:red;font-size:30px;">My name is Jim, I am a teacher in Tucson.</prose>
**Functionality:**

In the above example, we have changed three attributes of the text: font type, font color, and font size (the example calls for Arial font, red color, 30pt size).

We can also change one or more of these attributes in the *middle* of a sentence:

**Example:**

What if we want small some red text in the middle of the sentence?

<prose>What if we want <font size="8" color="red">some red text</font> in the middle of the sentence?</prose>

**M. Table <table>**

The `<table>` element adds a table to the notebook. The table may simply contain text, or can be fill in the blank, or a combination of both.

**Example:**

<table>
<thead>
<tr>
<th>Month</th>
<th>Savings</th>
</tr>
</thead>
<tbody>
<tr>
<td>January</td>
<td>$100</td>
</tr>
</tbody>
</table>

<table border="1">
  <tr>
    <td>Month</td>
    <td>Savings</td>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
</table>

**Functionality:**

In the above example, `<tr>` tag defines the row in the table and `<td>` defines column in the table. If we wanted to add another row, we would simply add another set of `<tr>` tags containing the appropriate content. Note that `<td>` is nested within `<tr>`.
border="1" gives the thickness of all the lines making up the table. If we increase this number, the lines will be thicker. If we don’t want lines for the table then define the border = "0".

Table can accept the input values to graph the table with given values or as a fill in the blank type of question as shown in the following example:

Example:

<table>
<thead>
<tr>
<th>Month</th>
<th>Savings</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table border="1">
  <tr>
    <td>Month</td>
    <td>Savings</td>
  </tr>
  <tr>
    <td><fillUp size="12"/></td>
    <td><fillUp size="12"/></td>
  </tr>
  <tr>
    <td><fillUp size="12"/></td>
    <td><fillUp size="12"/></td>
  </tr>
  <tr>
    <td><fillUp size="12"/></td>
    <td><fillUp size="12"/></td>
  </tr>
  <tr>
    <td><fillUp size="12"/></td>
    <td><fillUp size="12"/></td>
  </tr>
</table>

Functionality:

The tag <fillUp> is used here indicating that the user can input some text. The software can then use this response in a number of different ways. This tag can also be used for a fill-in-the-blank type question.
N. Image `<img src>`

The image element inserts an image into the notebook. In this image tag we can define left/ right/ center/ middle and also size (in points) of the image.

```html
<img src="/Users/jimmy/Desktop/angry.gif" align="left" width="32" height="32" />
```

Example:

![Image Example](/Users/jimmy/Desktop/angry.gif)

**Functionality:**

In the above example, `/users/jimmy/Desktop/angry.gif` indicates where the image is located on the computer (its file path). Without this information, the software doesn't know where the image is located!

*align="left"* specifies the image will be left justified. We could make it right justified by using *align="right"* or center it using *align="center"*. *width="32"* and *height="32"* indicate the size of the dimensions of the image.

O. Movie `<a href>=` 

The movie element inserts a movie into the notebook. The movie can be a link to a file on the web or to a file on the hard drive. In movie we link the website or location of the movie by using the `<a href>` tag.

**Example:**

Click here to see a movie!

```html
<a href="http://www.tizag.com/pic/flash/motiontweenle.swf">click here to see a movie!</a>
```

**Functionality:**
In the above example, a movie located on the web is linked. The text "Click here to see a movie!" appears on the notebook as clickable text. When this text is clicked, the movie begins playing.

Linking to a movie on the hard drive is similar, but with the filepath of the movie in place of a movie URL.

\[
\text{\textless a href="/users/jimmy/Desktop/motiontween1e.swf"}>\text{click here to see a movie!}</a>
\]

**P. Launch Apparatus <launchApparatus>**

The <launchApparatus> element launches (opens) an apparatus. The user can launch (open) an apparatus either by clicking its icon on the lab bench or within the lab notebook where prompted.

Example:

*** insert picture here of a prompt with clickable icon***

<launchApparatus Name="Social network">
  <dataFiles>
    <dataFile>
      <path>socialnetwork/basicgraphtheory.sn</path>
    </dataFile>
  </dataFiles>
  <prompt>Here is an example of a graph</prompt>
  <options>
    <statistics>show</statistics>
    <layout>Circle Layout</layout>
    <labels>Selected Vertex</labels>
  </options>
  <resultAction>
    <if result="true">
      <p>...</p>
    </if>
    <if result="false">
      <p>...</p>
    </if>
  </resultAction>
</launchApparatus>

**Functionality:**
In the above example, the lab gives a prompt `<prompt>` and a clickable icon. When the icon is clicked, the apparatus "Social Network" is launched (opened).

The Social Network apparatus opens a data file `<dataFile>` which is nested within `<dataFiles>`. (This organization may seem unnecessary in this particular example, since only one particular data file can be selected to open, but other times it will be useful to have the student select a data file amongst several data files). Each `<dataFile>` has a location given by `<path>`.

Each type of apparatus has its own set of options `<options>`. These are specific to each apparatus, so we won't go into detail here. See the guide for the relevant apparatus.

If more than one file is available, files can be accessed by a pull down menu showing all the titles `<title>` of the available files as shown below:

**Example:**

```xml
<launch Apparatus= "Social network" >
  <dataFiles>
    <dataFile>
      <path>socialnetwork/Rex.sn</path>
      <title>People Rex Knows</title>
      <description>This is Rex's social network.</description>
    </dataFile>
    <dataFile>
      <path>socialnetwork/Equation.sn</path>
      <title>Equations Example</title>
      <description>This network describes how variables are used in an equation.</description>
    </dataFile>
  </dataFiles>
  <prompt>Choose a network: </prompt>
  <options>
    <statistics>show</statistics>
    <layout>Circle Layout</layout>
    <labels>Selected Vertex</labels>
  </options>
</launch Apparatus>
```
**Functionality:**

In the above example, there is a pull-down menu where the user can select between two different files with titles: People Rex Knows and Equations Example. The user sees these files as titles `<titles>` since file names are not always descriptive. For example, the title: "People Rex Knows" is more descriptive than the file name: Rex.sn. Each file also has some small amount of text explaining what it is `<description>`. (maybe by right-clicking the pull down menu?)

In some cases, an apparatus can give feedback to the user about values that the user has entered into the apparatus. For example, a user might be asked to enter in some values into an apparatus and the apparatus checks those values and lets the user know if the values were correct or not.

**example:**

```xml
<launchApparatus Name="Social network">
  <dataFiles>
    <dataFile>
      <path>socialnetwork/basicgraphtheory.sn</path>
    </dataFile>
  </dataFiles>
  <prompt>Here is an example of a graph</prompt>
  <options>
    <statistics>show</statistics>
    <layout>Circle Layout</layout>
    <labels>Selected Vertex</labels>
  </options>
  <resultAction>
    <if result="true">
      <p>Correct!</p>
    </if>
    <if result="false">
      <p>Incorrect. Try again.</p>
    </if>
  </resultAction>
</launchApparatus>
```

**functionality:**

In the above example, the `<resultAction>` element is added which contains an `<if>` element with two possible attributes: "true" and "false". If the user enters the correct value into the apparatus, the apparatus sends back a message "Correct!" If the
user enters the wrong value, the apparatus sends a message back "Incorrect. Try again." These messages appear on the chalkboard and can be customized to say whatever the lab author wants.

**Q. Multiple Choice Question** `<multipleChoiceQuestion>`

The `<multipleChoiceQuestion>` element gives a question in multiple choice format. The following example shows the general format for a multiple choice question in XML. However, there are many possible options for these questions and their functionality is different depending on which section of the lab the question is located, as discussed in the sections to follow.

**Example (general):** (ignore yellow border)

```xml
<multipleChoiceQuestion>
  <question>1. Which beaker(s) contain(s) a solution that is hypertonic to the bag?
  
  ![Image of beakers]

  - a. Beaker 3
  - b. Beakers 2 and 4
  - c. Beakers 1, 2, and 5
  - d. Beaker 4
  - e. Beakers 3 and 4
</question>

<incorrectChoice>
  <choice>Beaker 3</choice>
</incorrectChoice>
<incorrectChoice>

```
<choice>Beakers 2 and 4</choice>
</incorrectChoice>
<correctChoice>
  <choice>Beakers 1, 2, and 5</choice>
</correctChoice>
<incorrectChoice>
  <choice>Beaker 4</choice>
</incorrectChoice>
<incorrectChoice>
  <choice>Beakers 3 and 2</choice>
</incorrectChoice>
</multipleChoiceQuestion>

**Functionality (general):**

All multiple choice questions must contain an actual question. The question is indicated by the tag `<question>`.
If desired, an image can also be displayed as part of a question, as in the example above. Several possible answers are then given indicated by the tag `<choice>`. Because multiple choice questions will be graded by the software, each `<choice>` must be contained within an element indicating if it is the correct or incorrect choice. These elements are `<incorrectChoice>` and `<correctChoice>`.

Students indicate their answer by clicking on a button next to the appropriate choice (each possible answer has its own button). The buttons are not filled in by default, and become solid black when clicked upon indicating that the student has selected that choice.

Beyond the general functionality explained above, multiple choice questions can have additional functionality depending on the element in which they are contained. `<multipleChoiceQuestion>` elements can be nested within the following elements which change their functionality (these are listed in the order in which they would appear in a lab):

- `<practiceQuestion>`
- `<preLabQuiz>`
- `<labSection>`
- `<labQuiz>`
- `<postLabSection>`
- `<openLabSection>`

Differences in multiple choice question functionality are discussed in the following sections:
1. `<multipleChoiceQuestion>` within `<practiceQuestion>`

Practice questions are multiple choice questions located within the prelab section. These questions give feedback in the form of grading as well as hints (if specified by the lab author).
Example:
*based on LabBench http://www.phschool.com/science/biology_place/labbench/

What the student initially sees (ignore yellow border):

![Practice Question](image1)

What the student sees upon selecting a wrong answer and clicking the “Check Answer” button.

![Practice Question](image2)

*Sorry. The correct answer is c. Hypertonic means that the solute concentration in the beaker is higher than in the bag. This is true for beakers 1, 2, and 5. Beaker 3 is isotonic to the bag, and beaker 4 is hypotonic to the bag.*
<b>Practice Question</b>

1. Which beaker(s) contain(s) a solution that is hypertonic to the bag?

To answer this question, compare the concentrations inside and outside of the bag.

The correct answer is c. Hypertonic means that the solute concentration in the beaker is higher than in the bag. This is true for beakers 1, 2, and 5. Beaker 3 is isotonic to the bag, and beaker 4 is hypotonic to the bag.

- Incorrect Choice: Beaker 3
  - Feedback: Incorrect.

- Incorrect Choice: Beakers 2 and 4
  - Feedback: Incorrect.

- Correct Choice: Beakers 1, 2, and 5
  - Feedback: Correct!

- Incorrect Choice: Beaker 4
  - Feedback: Incorrect.

- Incorrect Choice: Beakers 3 and 2
  - Feedback: Incorrect.
Functionality:

Each choice has a button that students can click. Once clicked, the button turns completely dark (illustration above shows a small dot in the circle, but that isn’t clear enough).

The user then clicks a “Check Answer” button. Box is smaller than illustrated above, but obvious, located at the bottom of the question. Once the Check Answer button is clicked, three things happen:

• Grade marks appear (either a red X or green check).

If the choice the user selected is incorrect, an explanation is given at the bottom of the question in RED lettering. Also, incorrect choices generate a red “X” to the left of the multiple choice answers (see fig.)

If the choice the user selected is correct, an explanation is given at the bottom of the question in GREEN lettering. Also, correct choices generate a green “check mark” to the left of the multiple choice answers (in the same location as the red “X” described above).

• Feedback is given.

In order to tell the user if their answer is right or wrong, the <feedback> element is used. The author of the lab can use whatever terms they want, in this example "Incorrect." and "Correct!" are used.

• An explanation is given directly following feedback.

In this example, there is a common explanation <explanation> that is given regardless of which choice the user selects. This explanation directly follows the feedback. If the lab author prefers to give a different explanation for each choice, not include the <explanation> tag. They would simply add the explanation to each choice's <feedback> area.

hint functionality: optional:

If the writer of the lab wants to include hints, there is a “Hint” box. It is located in the top right of the question. Students can click this button at any time (before or after attempting to answer the question).
The hint box icon looks like a tiny chalkboard. When the user clicks it, the Chalkboard (as described above) appears containing the hint (hint could be text, image, etc). The user can maximize, minimize, close, or move the window.

2. `<multipleChoiceQuestion>` within `<preLabQuiz>`

Example:
***insert a snapshot of quiz consisting of two questions, make text match below***

<!--note, an image is included with the second question-->

```xml
<br>PreLab Quiz. You must answer 2/3 questions correctly before going on to the Lab!</b></pre>
<prelabQuiz numberCorrectNeeded="2">
  <multipleChoiceQuestion>
    <question>1. How old are you?</question>
    <incorrectChoice>
      <choice>…</choice>
    </incorrectChoice>
    <correctChoice>
      <choice>…</choice>
    </correctChoice>
  </multipleChoiceQuestion>
  <multipleChoiceQuestion>
    <question>2. What is this?</question>
    <br/>
    <img src="5beakers.jpg"/>
    <incorrectChoice>
      <choice>…</choice>
    </incorrectChoice>
    <correctChoice>
      <choice>…</choice>
    </correctChoice>
  </multipleChoiceQuestion>
  <multipleChoiceQuestion>
    <question>3. How many?</question>
    <br/>
    <incorrectChoice>
      <choice>…</choice>
    </incorrectChoice>
    <correctChoice>
      <choice>…</choice>
    </correctChoice>
  </multipleChoiceQuestion>
</prelabQuizQuestion>
```
Functionality:

<multipleChoiceQuestion> elements within <preLabQuiz> elements function similarly to when they are within <practiceQuestion> elements with the following differences:

• There is a single Check Answers button which is clicked once all questions have been answered.

• When Check Answers is clicked, “CORRECT!” is displayed in green below each question that was answered correctly, “INCORRECT” is displayed in red below each question that was answered incorrectly.

• If there are enough correct answers to pass the quiz, a message appears saying: “CONGRATULATIONS! You passed the quiz. You may now begin the lab.”

• If quiz is not passed, (as indicated by the numberCorrectNeeded= attribute) a message saying: “You must answer 2/3 questions correctly before starting the lab.” A single button appears. When that box is clicked, the same quiz is given again anew. Quiz appearance will go back to how it looked the first time the user came to the quiz; previous answers are gone, and correct and incorrect answers no longer indicated.

hints optional:

If desired, hints can be included for prelab quiz questions. However, the Hint button is not present the first time the user attempts a quiz. After the user clicks the Retake Quiz button, there is a Hint button shown for each question. Aside from this difference, hint functionality is the same as described above for practice questions.

3. <multipleChoiceQuestion> within <labSection>

Multiple choice questions given in the lab section do not have hints, feedback, or grade marks. Student answers are recorded by the software and graded with the results included in an answer sheet. No specific example is needed here.

Example (general):

<labSection>
    <multipleChoiceQuestion></multipleChoiceQuestion>
</labSection>
4. `<multipleChoiceQuestion>` within `<labQuiz>`

Multiple choice questions given as a lab quiz have the same functionality as described above for multiple choice questions within the lab section. The only differences are there are many multiple choice questions in a row and the quiz is graded as a unit (e.g. 8/10 correct).

5. `<multipleChoiceQuestion>` within `<postLabSection>`

If desired, multiple choice questions can be written into the Post Lab section. These questions have very basic functionality: they behave like Lab Quiz questions, but are not graded or recorded by the software.

6. `<multipleChoiceQuestion>` within `<openLabSection>`

If desired, multiple choice questions can be written into the Open Lab section. These questions have very basic functionality: they behave like Lab Quiz questions, but are not graded or recorded by the software.

**R. Fill in the Blank `<fillInTheBlank>`**

Fill in the blank questions can be added to any section of a lab. Fill in the blank answers will either be graded by the teacher or ungraded. Included on the website will be a grading rubric with answers for all fill in the blank (and free response) questions.

**Example:**

There are ____ moons orbiting Saturn and 2 moons orbiting __________.

<fillInTheBlank>There are <fillUp size="5"/> moons orbiting Saturn and 2 moons orbiting __________ <fillUp size="12"/>.</fillInTheBlank>

**Functionality:**

Answers are typed into a space within the given question `<fillUp>`. The size of the blank (given as number of characters) is given by `size="5"` attribute.

Responses will be saved as part of a saved lab regardless of which sections the questions appear. However, only answers given in the Lab section are included in the Answer Sheet document.
S. Free Response  <freeResponse>

Free response questions can be added to any section of a lab. These questions require more writing by the student, perhaps even a paragraph or two. They may be included within the lab for all students to do, or added on to the end of the lab as a homework assignment or option in-class assignment (this is especially useful in case some students get finished with the lab early).

Example:

Compare and contrast human social networks with food web networks.

<freeResponse numberOfLines="10">Compare and contrast human social networks with food web networks.</freeResponse>

Functionality:

Answers are typed within the given number of lines indicated by the numberOfLines="10". Responses will be saved as part of a saved lab regardless of which sections the questions appear. However, only answers given in the Lab section are included in the Answer Sheet document. Included on the website, or within the software package will be a grading rubric with answers for all fill in the blank (and free response) questions.

Free response questions could also be given out as homework to be completed without a computer. Maybe they could be included as a separate worksheet.
**Question Type Comparison Chart:** This chart summarizes the major functionality differences between different categories of questions within a lab.

<table>
<thead>
<tr>
<th></th>
<th>Practice Questions</th>
<th>PreLab Quiz Questions</th>
<th>Questions in Lab section</th>
<th>Lab Quiz Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>hint button</strong></td>
<td><em>optional.</em> Can be clicked at any time.</td>
<td><em>optional.</em> Only present when retaking a quiz, not show the first time</td>
<td>None.</td>
<td>None.</td>
</tr>
<tr>
<td><strong>format</strong></td>
<td>multiple choice</td>
<td>multiple choice</td>
<td>multiple choice fill in the blank free response</td>
<td>multiple choice</td>
</tr>
<tr>
<td><strong>grading</strong></td>
<td>by software</td>
<td>by software</td>
<td>by software (multiple choice)</td>
<td>by software</td>
</tr>
<tr>
<td></td>
<td>correct answers given to students with explanation</td>
<td>responses marked correct or incorrect but correct answers not shown to students</td>
<td>by teacher (fill in the blank, free response: grading rubric included)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>*must pass to move on to the lab</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>*can be retaken (unlimited)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>responses saved to Answer Sheet</strong></td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

Questions that are answered in the PreLab section are not included in the Answer Sheet. This is because Practice Questions and PreLab Quizzes already give feedback to the student. Including students' responses to these questions on the Answer Sheet would make the document unnecessarily complicated, containing information the teacher would not normally need.

**T. Turn in Lab `<turnInLab>`**

The `<turnInLab>` element serves a few functions:
- signifies the end of the lab
- locks in all responses so they cannot be changed
- tells the software to grade any multiple choice lab questions
- generates a completed lab that is saved as a file
generates an Answer Sheet document for the teacher containing graded multiple choice question from the Lab section and ungraded student responses to fill in the blank and free response questions.

Functionality:

Responses to all questions can be changed at any time and will be saved automatically whenever a change is made. Therefore, individual questions do not need to be submitted in the lab section.

At the end of the lab, there is a TURN IN LAB button, which is the “final submit” button once the entire lab is completed. Upon clicking, a dialogue box opens with:

“Would you like to review your answers before turning in the lab? Once you turn in the lab, you CANNOT go back and change your answers!”

buttons:

RETURN TO LAB
COMPLETE TURN IN

If RETURN TO LAB is clicked, students are allowed to go back to any part of the lab and change answers. If COMPLETE TURN IN is clicked, the lab is locked (maybe a message like: “CONGRATULATIONS! You have completed the lab.”

A Completed Lab is now saved as a file.
IV. Generated Files

A. Saved Lab

As students go through the lab, their work is saved. Work is saved automatically when the student goes to the next page, or they can also save manually. Work is also saved when a student clicks the "Turn in Lab" button. When the saved lab is opened or printed, it looks just as the lab book looked to the student as they filled it out: All the included text, images, graphs, plus anything the student added is included.

B. Answer Sheet

This file contains only the answers to Lab Questions and Lab Quiz Questions. This sheet contains any software-graded portions and leaves the ungraded portions for the teacher to grade.

-where is this file located? Web only?
-need to give example for how an answer sheet would look

C. Grading Rubric

Each lab contains a grading rubric that teachers can use for fill in the blank and free response questions. This file will be located on the website. Only the following types of questions generate answers that sent to the Answer Sheet:

- Lab Questions (multiple choice, fill in the blank, free response)
- PostLab Quiz Questions (multiple choice)

V. Grading

Post lab quizzes will be multiple choice and graded by the software. The results for each student will be sent to the website and accessible by the teacher. The following stats will be given:

- grade for each student in points and percent (i.e.: 8/10, 80%)
- class average
- for each question, the number of students that answered incorrectly (not only is this information useful to the teacher, but it's useful to us as well…some questions may be too easy, too hard, or worded poorly)

If there are other multiple choice lab questions located before the post-lab quiz, these can be graded as well and included as a total score.
If there are fill in the blank questions and/or free response questions as well, these must be graded by the teacher. The teacher will then manually calculate a grade for each student weighting each section however they choose.