

Chemical Synthesis Automation

A Major Qualifying Project

Submitted to the Faculty of

Worcester Polytechnic Institute

in partial fulfillment of the requirements for the

Degree in Bachelor of Science

in

Computer Science

By

Elina Barrows

Lauren Fleming

Katherine Martin

Date: 4/6/21

Sponsoring Organization: AbbVie

Project Advisors:

Professor Therese M. Smith, Advisor

This report represents work of WPI undergraduate students submitted to the faculty as evidence of a degree requirement. WPI routinely publishes these reports on its web site without editorial or peer review. For more information about the projects program at WPI, see <http://www.wpi.edu/Academics/Projects>.

Abstract

The Chemical Synthesis Automation team continued the work of previous MQPs towards developing a robot platform that can automate steps of a chemical synthesis process. We examined the robot's equipment and reaction mechanisms to ideate an intuitive user interface by which chemists can input synthesis instructions. We designed a graphical LabView interface for manual input, and began the construction of an intermediate language to compile the input into a finite state machine.

Contents

Abstract	2
Table of Figures	4
Chapter 1: Introduction	5
1.1 Model Checking	5
Chapter 2: Examination of Equipment.....	7
2.1 Types of Equipment	7
2.2 Set Up	7
Chapter 3: LabVIEW Interface	13
3.1: LabVIEW Details	13
3.2 Natural Language Processing.....	16
3.3 Flow Charts as a Representation for State Transition Diagrams	19
Chapter 4: Program Ideation	24
4.1 Software Structure	24
4.2 Additional Tools	25
Chapter 5: Further Applications: CML and Chemical Tagger.....	27
5.1 Chemistry Markup Language (CML)	27
5.2 Chemical Tagger	28
Citations	30

Table of Figures

Figure 1: A graphical representation of a simple state machine.....	5
Figure 2: An example of a LabVIEW state machine.....	6
Figure 3: Block Diagram for LabVIEW Generate XML File.....	17
Figure 4: Block Diagram for LabVIEW Parse XML File	18
Figure 5: Simple Display of Reading vs Writing to XML in LabVIEW	2618
Figure 6: Flow Chart of the Synthesis of Acetanilide.....	2622
Figure 7: A Diagram Representing a Potential Structure of Components for the Software.....	2624
Figure 8: A Table Representing what the Calculator Interface might look like	2626
Figure 9: A Sample Entry in a CML File representing the lower bound of a temp profile	2627
Figure 10: Output of Running ChemicalTagger on the chemistry example above	2629

Chapter 1: Introduction

1.1 Model Checking

State machines, also called finite-state automata, are a method of encoding computer systems whose behavior depends on a series of inputs. They have a wide variety of uses, including specifying a program which responds to control commands and sensor readings (MIT OpenCourseWare). State machines consist of states, inputs, and transition functions. States are conditions that the machine can be in at any given point in time. They include an initial state, in which the machine starts, and final states, in which the machine is allowed to stop. Inputs, or events, are information received and processed by the machine. Transition functions determine how the machine reacts to receiving inputs. These functions generally change the machine from one state to another state, depending on where it starts and which input it receives (Myers).

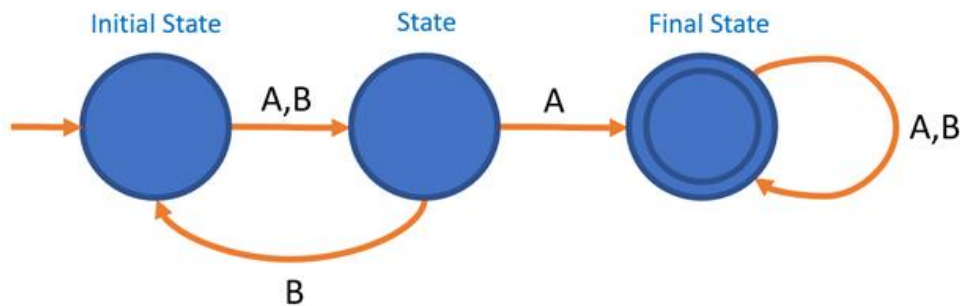


Figure 1: A graphical representation of a simple state machine. Blue circles represent the states, black text represents the inputs, and orange arrows represent the transition functions.

State machines are an important design element in LabVIEW. They are implemented using a case structure placed inside a while loop. The case structure contains all the different states of the machine, and its case selector processes the given input to choose the initial state. Inputs can be

either direct user input or programmed calculations, and the transition functions are handled by a shift register (LabViewWiki).

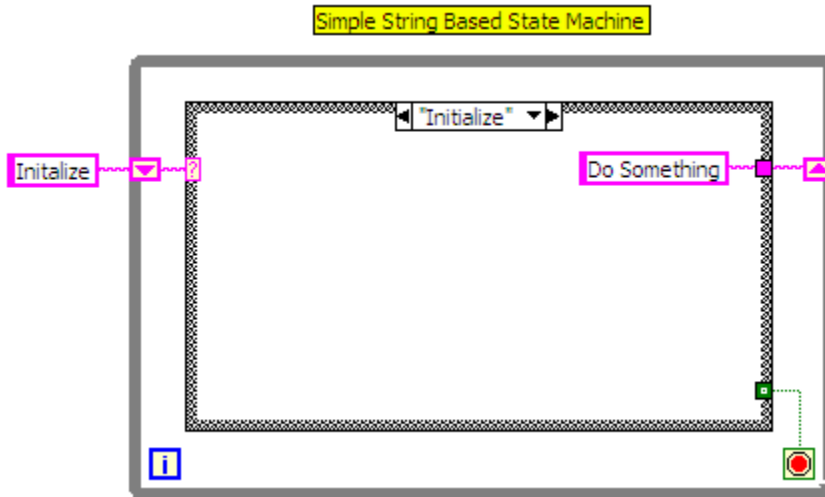


Figure 2: An example of a LabVIEW state machine (LabViewWiki). The outermost loop is the for loop, and the inner loop is the case structure, with a state called “Initialize”. The pink text boxes and wiring represent the inputs and transition functions.

Model checking is an algorithm to determine whether a state machine system behaves as intended. It requires defining a set of properties which indicate the correctness. For instance, model checking for a microwave might specify the property that the heat should not turn on unless the door is closed. These properties must then be expressed as logical statements (Schneider). Then, the state machine is simulated with various inputs, and the property statements are evaluated at each state the machine reaches. If the statements hold true along every transition path, then the machine obeys the properties correctly. If a statement evaluates as false at any point, then the machine can provide a counterexample to show that there is a problem where the machine does not work as intended and must be debugged (Sinha).

Chapter 2: Examination of Equipment

2.1 Types of Equipment

A list of potential equipment for current and future compatibility with the robot was comprised of both common equipment used during synthesis and from lists given from our sponsor at AbbVie. For each piece of equipment, a brief synopsis of its purpose was given along with a proposed action that the robot could direct it to do. The equipment was split up into four categories based on the portion of the synthesis it would be used for. The first section, “Set Up”, entailed devices that were used to prepare the reactants for synthesis, including measurements devices and glassware. The second section included devices that aid in the reaction portion of the synthesis such as heating and stirring. The third section includes devices used to isolate the product of the synthesis such as evaporation. The fourth and final section of the equipment list covers analytical equipment used to determine the chemical composition of the product, such as chromatography or spectroscopy. This MQP project will focus mainly on set up and reaction mechanisms while giving recommendations for future applications for product isolation and analytical tests.

2.2 Set Up

Measurement

- o Syringe Pump
 - § Details

- Depresses the syringe at a set rate over a given time (ie. 10 ml/min, 1 mL/hr)

§ Actions

- Depress to X Volume (or height)
 - o $X \text{ Volume} / \text{Rate} = \text{Amount of Time}$
- Pull Back to X Volume (or height)
- expand

o Scale¹

§ Details

- Scale requires both doors to close, has much more precision than the syringe pump

· Glassware²

o See footnote

React

· Heating

o Heating Mantle

§ Details

- Almost glove shaped, sometimes sand is used to fill in gaps to maximize surface cover

§ Actions

- Heat to Setting X

¹ Much like in baking, sometimes mass is a better indicator than volume. This is because reaction quantities are calculated by moles (or mass/molar mass). Therefore, the volume will accumulate more error because it is a derived measurement. In general, a scale will have more precise measurements (think +/- 0.001 g vs +/- 1 mL)

² There are many potential glassware setups for different kinds of chemical reaction types.

- o Rotate dial to Setting X
- o Heat to X
- o Once Temperature Reaches X, Turn Off

o Hot Plate/ Combination Hot Plate Stirrer

§ Details

- Can either have just hot plate or have stirrer attached

§ Actions

- Rotate dial to Setting X
- Heat to X
- Once Temperature Reaches X, Turn Off

o Bunsen Burner

§ Details

- Rarely Used

· Stirring

o Stir Plate and Magnetic Stir Bar/ Combination Hot Plate Stirrer

§ Details

- Can either have just hot plate or have stirrer attached, magnetic stir bar (size of a black bean) goes inside the flask and magnetic stir plate propels it. For this reason, flask needs to be touching or very close to stir plate

§ Actions

- Rotate dial to Setting X
- Stir for X minutes then Turn Off

- Reflux

- Details

- § Need to talk to AbbVie to see if this is necessary

Isolate Product

Distillation

- Fractional Distillation

Evaporation

- Rotovap

- Details

- Product is in solution and solution needs to be evaporated off.

- Solution is put in the pear-shaped flask (the one at an angle) and rotated while it is dropped into a hot water bath

- Actions

- Set rotation to setting/dial X
 - Set water bath to X temperature
 - Lower/raise arm

Separation

- Separation Funnel

- Details

- It seems like last year's group had a system in place already to do this

Vacuum Filtration

- Details

- Special glassware with a filter on top and vacuum suction beneath. The solution is pulled through the filter and the precipitate (aka the desired product) is left on the filter and is dried. This may need to be repeated to maximize product yield.
 - Usually before this step, the solution is cooled in an ice bath in order to help crystals form. If you don't see crystals, it may need to be cooled more.
- Actions
 - Turn on vacuum/turn on air valve
 - The air valve is usually found on the outside of the fume hood and is attached to the glassware by tygon tubing (small rubber hosing)
 - Dispense X solution onto the filter paper

Analytical Tests

Chromatography

- Gas Chromatography (GC)
- High Pressure Liquid Chromatography (HPLC)
- Thin Layer Chromatography (TLC)
- Gravity Columns/Column Chromatography
- Silica Gel Chromatography

Spectroscopy

- Infrared Spectroscopy (IR)
- Ultraviolet Spectroscopy (UV Vis)
- Nuclear Magnetic Resonance (NMR)

- H NMR
- C NMR

Physical Characteristics

- Melting Point Analysis
- Water Content
 - Karl Fischer Titration (KFC)
 - Details
 - Determines the water content of a solution by titrating it automatically. The volume and solution type are input before hand so the calculations are also automatic
 - Actions
 - Set solution type to X
 - Set volume to X
 - Dispense X mL of solutions
 - Send results to server

Chapter 3: LabVIEW Interface

3.1: LabVIEW Details

Consider the sample list of chemical synthesis steps provided below:

Quench Bot Chemical Language/Sequence

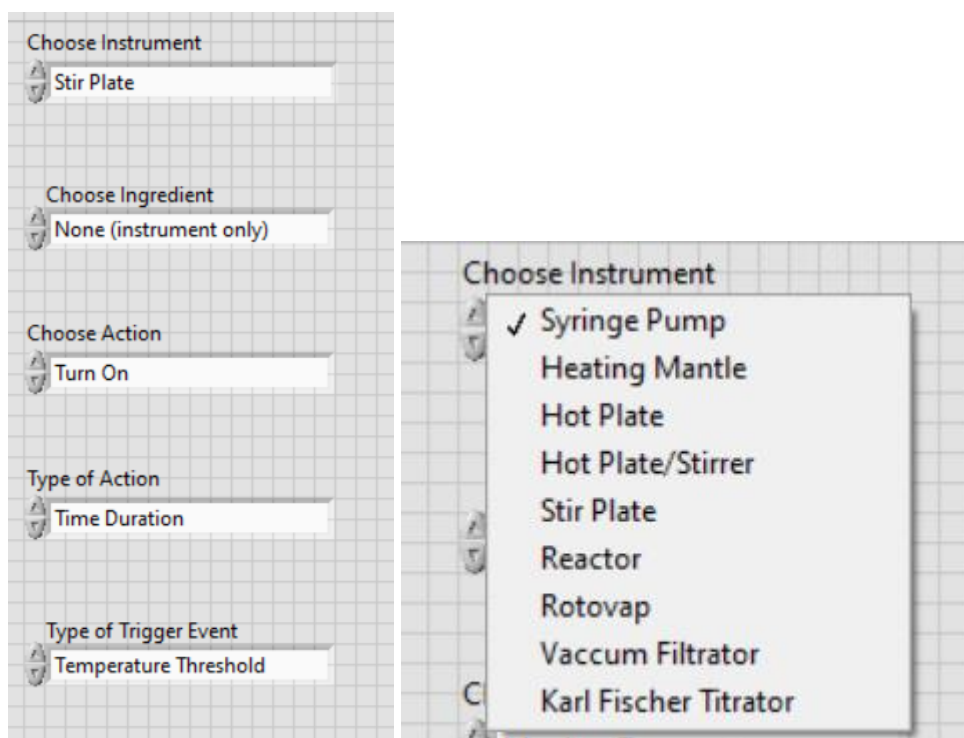
1. Turn on stir plate (300 rpm)
2. Charge syringe 1 (100 mL) with reactive reagent
3. Charge syringe 2 (100 mL) with quenching solution
4. Dose syringe 2 into reactor 10 mL/min
5. Stop dosing syringe 2 after 5 mins (50 mL)
6. Charge syringe 2 (50 mL) with quenching solution
7. Dose syringe 1 and 2 simultaneously into reactor at 10 mL/min
8. Monitor temp of reactor
9. If temp exceeds 35°C stop dosing syringe 1 and 2
10. Monitor temp of reactor
11. Once temp of reactor reaches 25°C start dosing syringe 1 and 2 at 10 mL/min
12. Continue until syringe 1 and 2 have completely dispensed
13. Fill syringe 1 (100 mL) with cleaning solvent (methanol)
14. Dose syringe 1 into reactor (50 mL/min) (2 mins)

15. Fill syringe 2 (100 mL) with cleaning solvent (water)

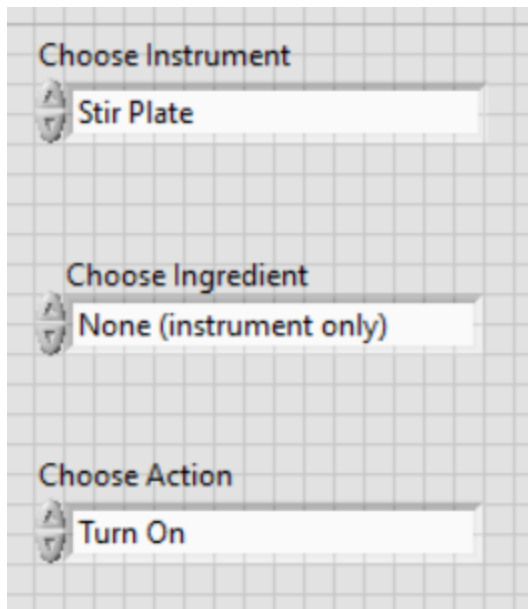
16. Dose syringe 2 into reactor (50 mL/min) (2 mins)

17. Turn stir plate off.

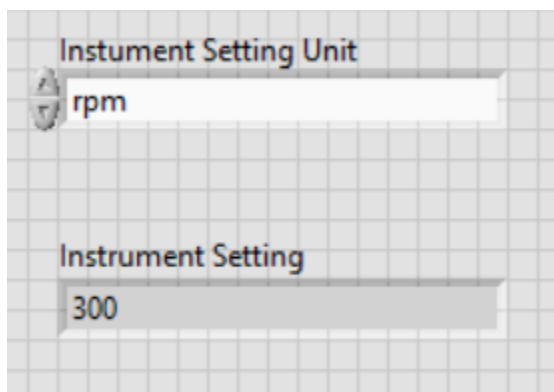
In order for the robot to carry out a synthesis such as this, chemists must be able to enter these steps in a manner that the robot can understand. A LabVIEW interface makes this possible.



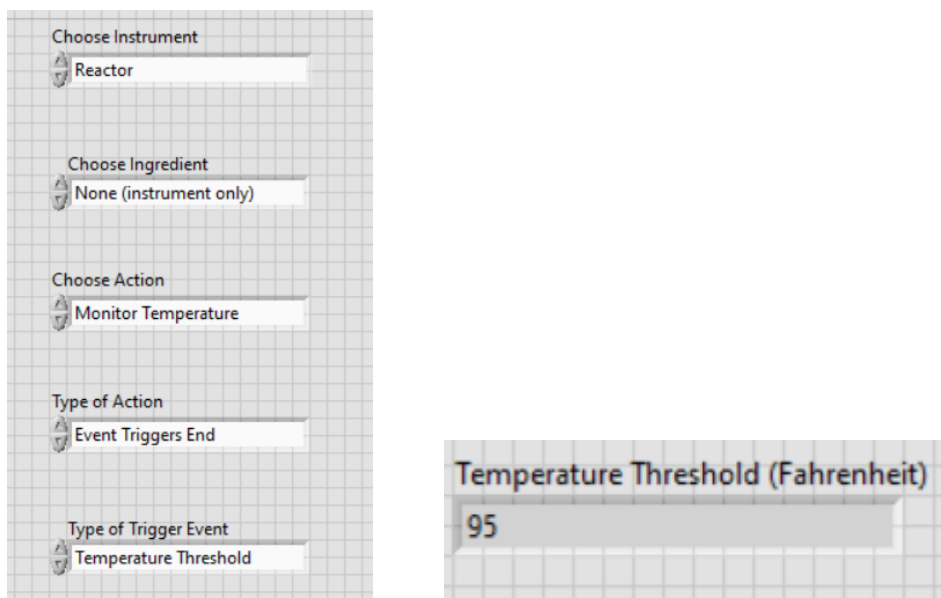
The drop-down menus and boxes above help chemists enter the details of each individual synthesis step. Consider step 1 of the example synthesis found above - “turn on stir plate (300 rpm)”.



The selections above instruct the robot to focus on the stir plate and complete the “turn on” action with no ingredients involved, since the plate must only be turned on. The user is then prompted to specify the setting that the instrument is to be set to.



These menus specify that the stir plate should be set to operate at 300 RPM, as said in the instructions. However, consider a more complicated instruction that requires a stop condition, such as example step 9, which asks for the dosing to stop when a certain temperature is reached.



The instructions specify that this step has a stop condition. In the previous step, a switch just had to be flipped and the plate turned on. In this case, the robot must continuously monitor the temperature and trigger a new event when the threshold is released. This command tells the system to send a notification when the temperature reaches the threshold and move to the next step. The system is also prepared to deal with other stop conditions such as time limits.

3.2 Natural Language Processing

The LabVIEW interface still leaves a lot of technical details up to the chemist. In order to make their job easier and stay in a language that they completely understand, natural language processing aspects were added to this project. Four separate LabVIEW interfaces were created and connected that can read, write, generate, and parse XML files. This breaches a significant gap between chemist and robot, since a chemist would typically struggle to write complex chemical reactions in terms that a robot would be able to understand. One of the major goals of this project was to limit mistakes in synthesis, and making a chemist write in computer language

is certainly leaving room for mistakes. The chemist's instructions can be run through a program known as Chemical Tagger, described in detail below. Chemical Tagger outputs an XML file containing the information necessary for a reaction to occur. The LabVIEW interfaces are able to convert these instructions into an understandable format.

The Easy Generate file can convert LabVIEW Data (such as a Boolean, String, or Integer) into an XML string with the LabVIEW data labels converted to XML item names and the LabVIEW data values converted into XML item values.

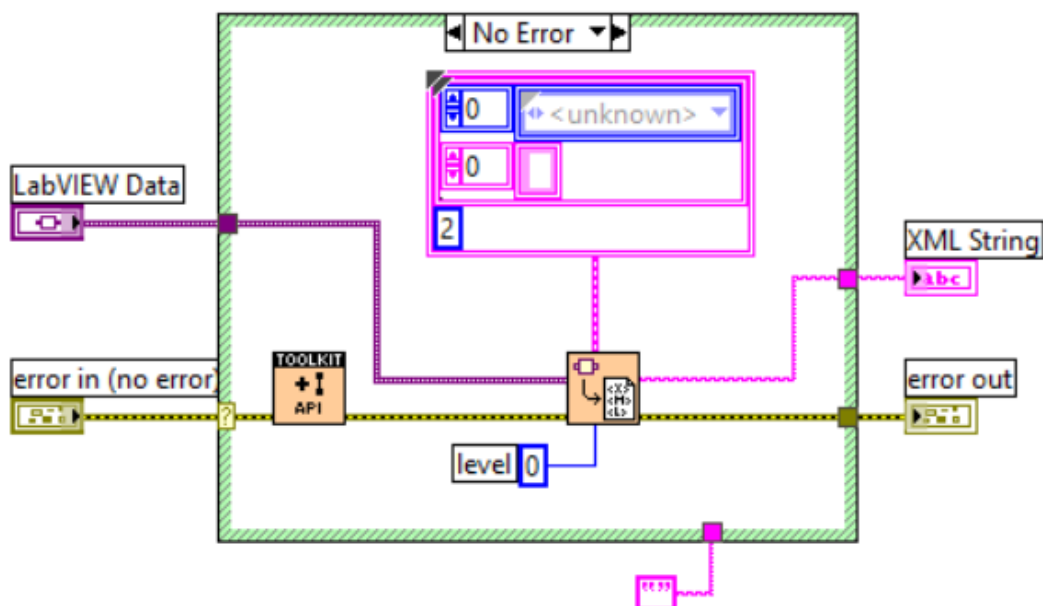


Figure 3: Block diagram for the LabVIEW Generate XML file.

The Parse XML file reads an XML file and converts the strings contained within the file to LabVIEW data, most commonly Strings and Integers, based on the included argument LabVIEW data (Type).

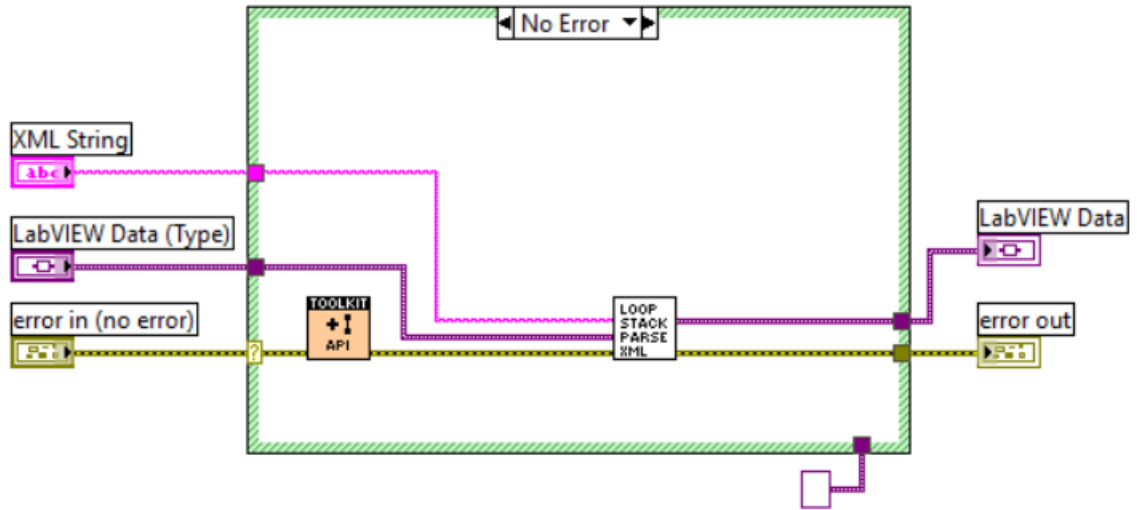


Figure 4: Block Diagram for Parse XML File.

The write and read files work essentially in opposite of each other. Writing to an XML file requires the conversion of LabVIEW data to an XML String and writing to an XML file with this. Reading to an XML file requires the conversion of XML Strings into LabVIEW data, the type of which is specified as an input.

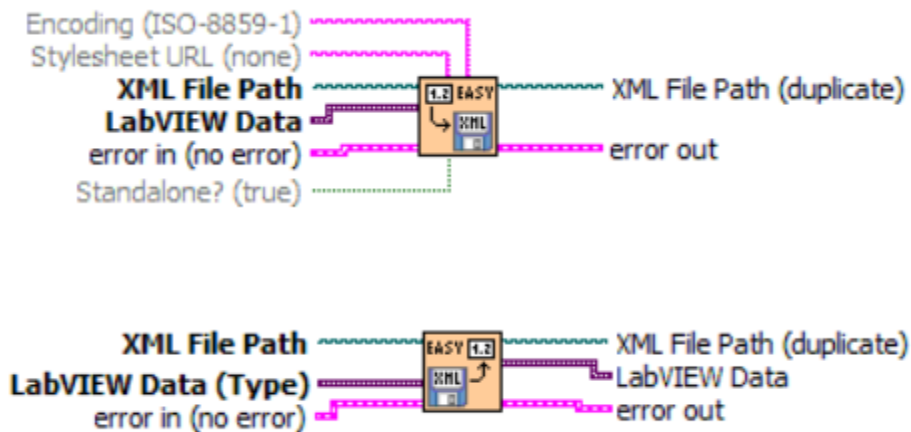


Figure 5: Simple Display of Reading vs Writing to XML In LabVIEW.

3.3 Flow Charts as a Representation for State Transition Diagrams

In order to represent a state diagram from a chemist's perspective, we first started with a flow chart. As an example, we used the synthesis of acetanilide, a compound formerly used as a non-prescription drug to treat pain and fevers and has a very similar structure to a more common drug, acetaminophen, which is Tylenol.³ We took synthesis instruction for this compound and transferred them to an intermediate flow chart form which could then be better understood by both the chemists and the computer scientist and robotic engineers working on the robot so that the chemists' language could be translated easier to the computer language. This also gave the group a better idea of how a common user would expect a state transition diagram to function within the robot. After thoroughly understanding the process itself and the many checks and balances that need to be put in place to ensure proper synthesis, we then plan to transfer the flow chart into a state transition diagram. The synthesis was transcribed into steps as follows.

Synthesis of Acetanilide

1. Add 7.0 mmol of aniline to 15 mL of deionized water in a 50. mL Erlenmeyer flask
2. Swirl the mixture thoroughly.
3. Add 8.3 mmol of acetic anhydride to the mixture.
4. Swirl the mixture for 5 minutes
5. Cool in an ice bath until crystal formation is visible. If no crystals form, it may be necessary to scrape the bottom of the flask.

³The difference between acetanilide and acetaminophen is the presence of an alcohol (-OH) group para to the amine group on acetaminophen.

6. Collect product for vacuum filtration. Product should be visible on the filter paper. If desired yield is not met or if product is still visible in the dissolved portion of the filtration, repeat filtration.
7. Add the crude acetanilide and 5 mL of water to a 25 mL beaker. If the solution does not dissolve, continue to add water dropwise until it is just dissolved.
8. Add additional water equal to 10% of the volume of the dissolved crude acetanilide.⁴
9. Add enough fine powder charcoal to so that the solution is opaque, even when swirled.⁵
10. Heat the solution until it boils while continuously stirring.
11. Prepare the beaker with a small amount of boiled water (0.5 cm) and a glass filtration funnel affixed with a fluted filter paper. The beaker should be warm so that the shock of adding hot solution does not break the glassware
12. Using gravity filtration, pour small amount of the charcoal-acetanilide solution through the filter paper at a time. Keep the solution hot in between pours by returning it to the heat. If the solution starts to recrystallize before filtration or on the filter paper, reheat and filter again.
13. Add boiling water (2 mL) to the flask containing the charcoal solution to wash the remaining solution through the filter paper. Additionally, add boiling water to the filter paper dropwise to filter any remaining acetanilide. If any decolorized carbon remains in the filtered product, repeat filtration back to Step 10, adding more water if necessary and by using a new clean filter paper.
14. Remove any additional water by boiling the solution.

⁴ For example, if after dissolved, the solution is 10 mL, add 1 mL additional water

⁵ This step is to remove any impurities present after the initial synthesis, as well as to remove any color present in the solid

15. Once concentrated, remove the solution from heat and allow it to. Cool to room temperature. Then place on ice. If no crystallization occurs, you may need to scrape the bottom of the glassware. If the crystallized product still looks impure (i.e., color or texture) Repeat recrystallization.
16. Collect product by vacuum filtration, and dry overnight.
17. Analyze the final product by testing its melting point.⁶

These steps were then transformed into a flow chart. For ease of viewing, the exact measurements were removed, and only basic commands are visible on the chart and the chart assumes that set up has already been completed. The chart works by including color coded action boxes driven and conditional arrows.

⁶The melting point will be depressed if water is still present in the product.

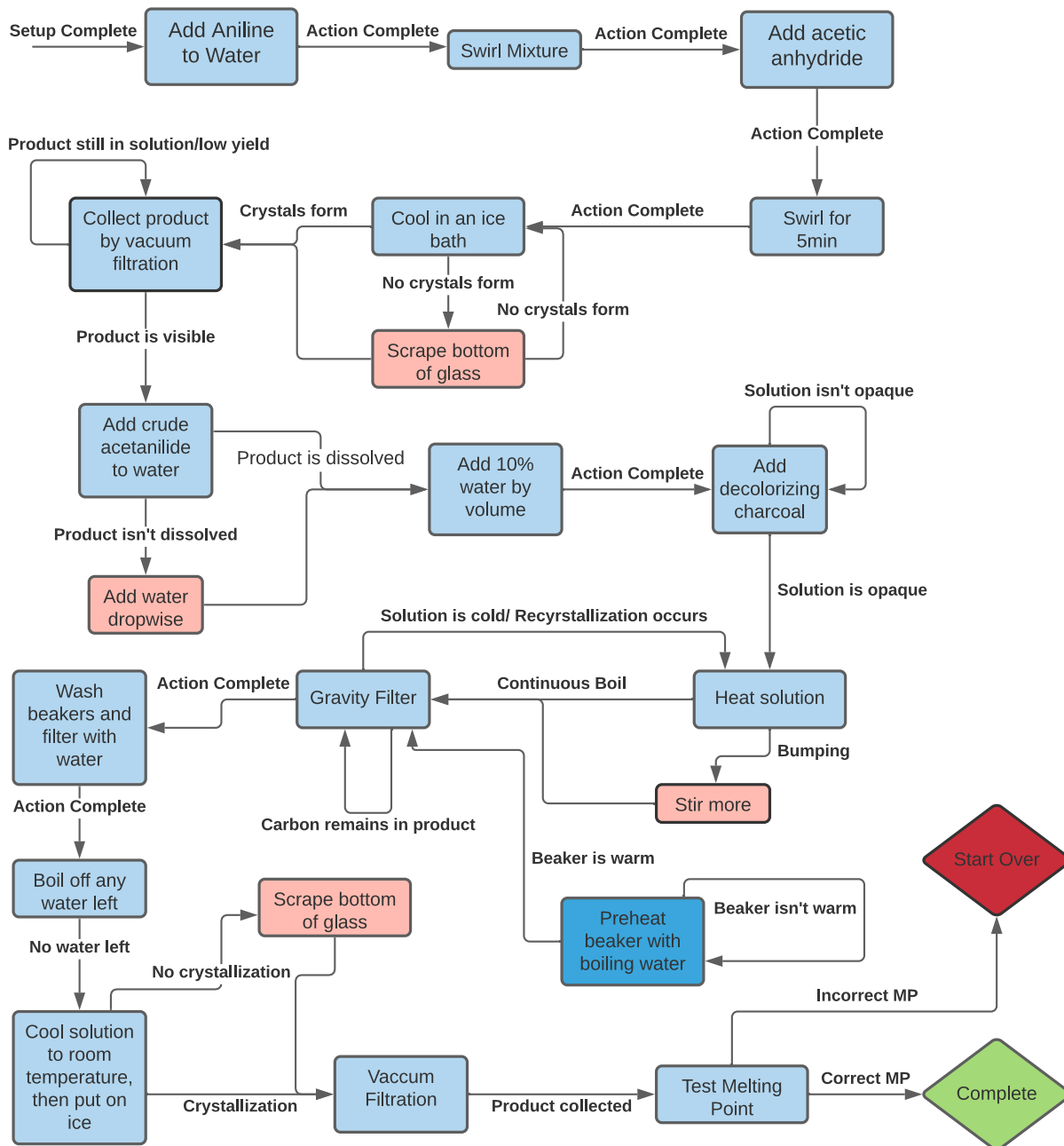


Figure 6: Flow Chart of the Synthesis of Acetanilide

The light blue boxes represent actions that drive the synthesis such as adding, heating, mixing. These would be the actions the robot would instruct each individual machine to do. The arrows represent the checkpoints that the robot, or user, will be looking for before completing the

next action. These include time (i.e., Stir for five minutes), temperature (i.e., heat to 30 degrees Celsius), and qualitative properties like light, color, opacity, odor, texture, etc. The red boxes represent situations where the arrow conditions are not meant, and the synthesis must deviate from the original path in order to get back on the right path. The dark blue box represents an action that is done concurrently alongside the main synthesis. This may be something that is required for the synthesis but does not need to be completed in order like actions in light blue boxes. Lastly, the diamonds represent the outcome of the synthesis, green for a successful synthesis of the product after analytical testing, and red for errors in synthesis. The red error may not require the user to completely start over but will require human intervention to decide what to do from there.

Chapter 4: Program Ideation

4.1 Software Structure

The chemical synthesis robot will need software to interpret a chemical recipe into instructions that the robot can process. We worked on defining a potential base structure for this software, which could be implemented in future work.

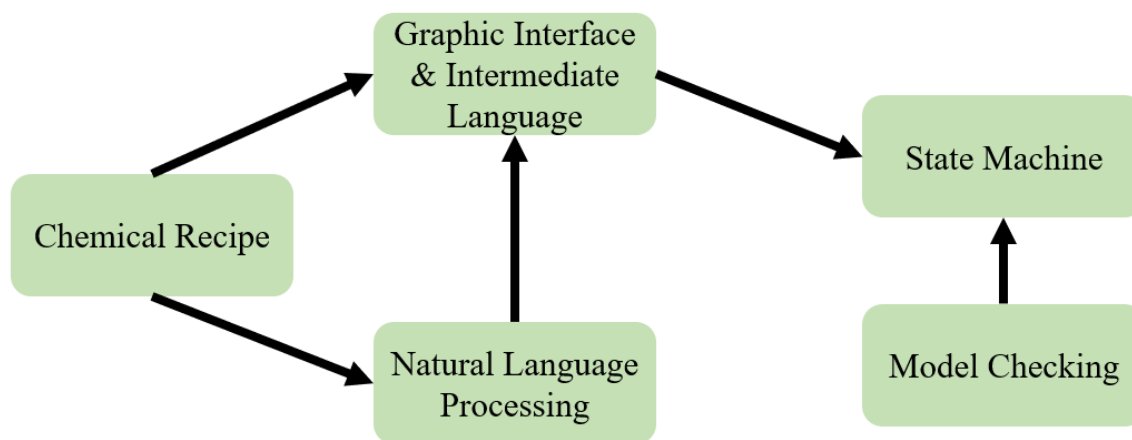


Figure 7: A diagram representing a potential structure of components for the software.

Figure 4 shows a diagram of what this program might look like. The first piece is a chemical recipe, the directions for the synthesis process as written by chemists in natural language.

This recipe must then be converted into a computer-readable form. One way to do this would be through a graphical user interface in LabView. We discussed this interface earlier, under “LabView Details”. It uses dropdown menus to allow chemists to select and input commands, and is designed to be intuitive even without coding experience. A compiler would

then convert this graphic program into a specially constructed intermediate language. This intermediate language would use a Lisp-style parenthesis notation, with a set of verbs describing the actions that can be taken by the synthesis robot. It would be designed to be relatively human-readable, but also easily interpretable into a state machine. As an alternative to using the graphical interface, a chemist could learn the intermediate language and code with that directly.

Another input route would be to use natural language processing to automatically interpret the written chemical recipe into the intermediate language form. This could then potentially employ the graphical interface to display the interpreted instructions, allowing chemists to double-check the results and edit if necessary.

The intermediate language instructions, which are ultimately obtained from any of these input methods, must then be compiled into a state machine. This machine contains the final code which would be used to tell the robot how to carry out the synthesis process. Finally, before these state machine instructions are ever executed, the software would employ model checking to ensure that the state machine is correct.

4.2 Additional Tools

One tool which we believe would be useful to implement in future work is a reaction calculator, which would allow chemists to specify a desired amount of product or limiting reactant, and from that compute how much of the other reaction components would be needed or produced. For instance, if using the equation for salt $\text{Na} + \text{Cl} \rightarrow \text{NaCl}$, the chemist could input the desired amount of sodium chloride to be produced and learn from the calculator how much sodium and chlorine would be required to make it. We envision that the quantities could be specified in grams or molar. The calculator would need to have access to a database which

holds the molecular weights of each chemical; beyond that it would just use a set of equations, so we think this would be very possible to implement.

	Na	+	Cl	→	NaCl
g					
M					

Figure 8: A table representing what the calculator interface might look like. Entering a quantity in one of the white boxes on the table would calculate the quantities for each of the other boxes.

Chapter 5: Further Applications: CML and Chemical

Tagger

5.1 Chemistry Markup Language (CML)

Chemistry Markup Language, abbreviated as CML, is an implementation of XML used to manage molecular information. We are able to specify an acceptable range of parameters of a reaction so that the computer can influence reaction rates to control the behavior resulting from the synthesis steps outlined by the chemist. One feedback system for such a project is temperature monitoring with the use of a temperature profile, or how the temperature of the reaction changes as a function of distance or time. Including such parameters with CML ensures that our reactions are maintained how they need to be and may further remove human involvement from the chemical synthesis steps.

```
<entry id="lowerBound" term="Lower Temperature Bound"
      dataType="xsd:double" unitType="unitType:amount" units="unit.dalton">
  <!--Datatype double will be replaced by a list of ordered pairs containing temps and times.-->
  <definition>
    <xhtml:p>
      This is where the definition of the entry will go.
      The lower bound of a temperature profile is the lowest temperature
      that the reaction will be able to be at.
    </xhtml:p>
  </definition>
  <description>
    <xhtml:p>
      Further description of the item will go here if necessary
    </xhtml:p>
  </description>
</entry>
```

Figure 9: A Sample Entry in a CML File Representing the Lower Bound of a Temperature Profile.

5.2 Chemical Tagger

ChemicalTagger is a parser used as a tool for text-mining within chemistry. It is classified as medium-depth and phrase-based and is optimized to be used on the language of chemical reactions and chemical experiments (Hawizy). Our LabVIEW files that are capable of reading and writing XML take an XML file as output from ChemicalTagger after it has ran on natural language provided by a chemist. These XML files contain all of the necessary information about the desired reaction in the correct format. ChemicalTagger uses three separate taggers to recognize various parts of the input sentences (Hawizy). OSCAR4 is used to detect chemical entities, Regex for detecting key words, and OpenNLP for finding English parts of speech that do not fit these other categories, such as prepositions (Hawizy).

Consider the following text as being input from a chemist who wishes to use the chemical synthesis robot:

To a stirred solution of 4-hydroxypiperidine (0.97 g, 9.60 mmol) in anhydrous dimethylformamide (20 mL) at 0°C was added 1-(bromomethyl)-4-methoxybenzene (1.93 g, 9.60 mmol) and triethylamine (2.16 g, 21.4 mmol). The reaction mixture was then warmed to room temperature and stirred overnight. After this time the mixture was concentrated under reduced pressure and the resulting residue was dissolved in ethyl acetate (40 mL), washed with water (20 mL) and brine (20 mL) before being dried over sodium sulfate. The drying agent was filtered off and the filtrate concentrated under reduced pressure. The residue obtained was purified by flash chromatography (silica gel, 0-5% methanol/methylene chloride) to afford 1-(4-methoxybenzyl)piperidin-4-ol as a brown oil (1.70 g, 80%).

Running ChemicalTagger on this language produces the following output:

To a stirred solution of 4-hydroxypiperidine (0.97 g , 9.60 mmol) in anhydrous dimethylformamide (20 mL) at 0 °C was added 1-(bromomethyl)-4-methoxybenzene (1.93 g , 9.60 mmol) and triethylamine (2.16 g , 21.4 mmol) . The reaction mixture was then warmed to room temperature and stirred overnight . After this time the mixture was concentrated under reduced pressure and the resulting residue was dissolved in ethyl acetate (40 mL) , washed with water (20 mL) and brine (20 mL) before being dried over sodium sulfate . The drying agent was filtered off and the filtrate concentrated under reduced pressure . The residue obtained was purified by flash chromatography (silica gel , 0-5 % methanol / methylene chloride) to afford 1-(4-methoxybenzyl)piperidin-4-ol as a brown oil (1.70 g , 80 %) .

Actions:
 Add Heat Dissolve Filter Wash Purify Dry Yield
 Concentrate Stir

Figure 10: Output of running ChemicalTagger on the chemistry example above. Sourced from <http://chemicaltagger.ch.cam.ac.uk/submit>.

ChemicalTagger is capable of separating the text into various reaction steps accompanied by actions, just as a robot would need in order to carry out the synthesis. This intermediate step provides the perfect gateway between chemist and robot. Further implementations of this project may wish to optimize ChemicalTagger to perfectly fit the chemical language and terminology that the robot must understand. One could even aspire to create a seamless transition between LabVIEW and ChemicalTagger without the need to separately transform the plain language into an XML file before being sent to LabVIEW.

Citations

LabViewWiki: State Machine. (2020, June 3). Retrieved December 2, 2020, from

https://labviewwiki.org/wiki/State_Machine

MIT OpenCourseWare: Chapter 4, State Machines. (2011, April 25). Retrieved December 2,

2020, from https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-01sc-introduction-to-electrical-engineering-and-computer-science-i-spring-2011/unit-1-software-engineering/state-machines/MIT6_01SCS11_chap04.pdf

Myers, A. (2006, May 1). CS 211 Spring 2006 State Machines. Retrieved December 3, 2020,

from http://www.cs.cornell.edu/courses/cs211/2006sp/Lectures/L26-MoreGraphs/state_mach.html

Schneider, G. (n.d.). Model Checking: A Complement to Test and Simulation. Retrieved

December 2, 2020, from <http://www.cse.chalmers.se/~gersch/model-checking/What-is-Model-Checking.html>

Sinha, N. (n.d.). Model Checking Guided Tour. Retrieved December 2, 2020, from

<http://www.cs.cmu.edu/~modelcheck/tour.htm>

Hawizy, L. (n.d.). ChemicalTagger: A tool for semantic text-mining in chemistry. Retrieved May

4, 2021, from <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3117806/>