Unit Verification and Validation Plan for Stoichiometry Mass-Mass Program

Deemah Alomair

December 24, 2019

1 Revision History

Date	Version	Notes
24/12/2019	1.0	First version of document

Contents

2 Symbols, Abbreviations and Acronyms

symbol	description
SRS	Software Requirements Specification
SMMP	Stoichiometry Mass-Mass Program
T	Test

3 General Information

This section gives an overview of purpose of this document. In addition, describe the scope of the unit testing that will be carried on throughout the testing phase.

3.1 Purpose

The main purpose of this document is to build verification and validation plan for Stoichiometry Mass-Mass Program, provide description about the unit testing that will be carried out on SMMP, and check whether each SMMP function meets the SRS and fulfill its intended purpose. This document will be used as the reference and guidance for testing SMMP.

3.2 Scope

- The tests outlined in this document are limited to the verification of the requirements given in the SMMP SRS.
- The tests outlined in this document will include dynamic tests.
- The tests outlined in this document will include manual and automated tests.
- The tests outlined in this document will include integration testing.
- The tests outlined in this document will include parallel testing in limited basis.
- The SMMP software will be written in Python. The testing of implementations in other languages will not be considered in this document.

4 Plan

4.1 Verification and Validation Team

The test team includes the following members:

- Main contributor: Deemah Alomair.

- Primary reviewers: Dr. Spencer Smith, Sharon Wu.
- Secondary reviewers: Bo Cao, Ao Dong, Peter Michalski.

4.2 Automated Testing and Verification Tools

The automated testing for SMMP will be carried out using multiple testing frameworks. Unit testing , coverage testing and linters will be considered. Throughout the implementation of SMMP I will use Pylint package. It is one kind of linters that is capable of highlighting, underlining, or otherwise identifying problem areas in the code before I run it. So, It Checks for errors, tries to enforce a coding standard. Also, it is considered as Logical, and Stylistic tool for python. Moreover, PyCharm environment provides different types of preinstalled testing options like unit testing. For coverage testing coverage package will be used.

4.3 Non-Testing Based Verification

Not Applicable.

5 Unit Test Description

This unit test description has been built to summarize all the unit tests that will be conducted in testing phase for each module individual functions. all modules related functions can be found in MIS document .

5.1 Tests for Functional Requirements

5.1.1 input conversion test

First module will cover all individual functions in the first part of the SMMP which is converting user input into chemical reaction parameter..

1. **T1: get number of chemical reaction elements**

   Type: Functional, dynamic, automatic.

   Initial State: Not Applicable.

Input: user must input at least two chemical reactants and one product

input	example	output
one reactant and products	$H_2O \rightarrow H_2 + O_2$	error massage
two reactant and no product	$H_2 + O_2 \rightarrow$	error massage
two reactant and one product	$H_2 + O_2 \rightarrow H_2O$	accepted
two reactant and two product	$FeCl_3 + MgO \rightarrow Fe_2O_3 + MgCl_2$	accepted

Table 1: Input possible entries and its corresponding outputs

Test Case Derivation: the expected entry is chemical reaction with at least two reactants and one product.

How test will be performed:

- the user will choose an elements from drop down menus using GUI widget for each reactant and product
- the system will check if user choose at least one element for reactant 1, reactant 2 and product 1 to processed.
- if user did not fulfill the minimum requirement system will show up an error massage indicating that.

2. **T2: get name of chemical reaction elements**

Type: Functional, dynamic, automatic.

Initial State: Not Applicable.

Input: user must input an element in both side of chemical reaction.

input	output	explenation
$H_2 + O_2 \rightarrow H_2O$	accepted	H , O in both side of the reaction
$FeCl_3 + MgO \rightarrow Fe_2O_3$	error massage	Mg is in reactant side but not in product side
$CO_2 + O \rightarrow CH_4 + O_2$	error massage	H is in product side but not reactant side

Table 2: Input possible entries and its corresponding outputs

Test Case Derivation: the expected entry is consistent chemical reaction where each element entered the reaction must produced from the reaction.

How test will be performed:

- the user will enter an elements from drop down menus using GUI widget for each reactant and product
- the system will check if user enter the element in both sides of the reaction to processed.
- if user did not fulfill the minimum requirement system will show up an error massage indicating that.

3. **T3: get atom value of chemical reaction elements**

Control: Functional, dynamic, automatic.

Initial State: atom value = 0 if no element has been selected.

Input: atom value:

atom value	response
atom value $\notin$ number	error massage
atom value ≤ 0	error massage
atom value > 0	accepted

Table 3: Input possible value's level

Output: error massage indicating that atom value is not a number or atom value ≤ 0.

Test Case Derivation: the expected value is chemical element selected from a menu and atom value belongs to positive integer number.

How test will be performed:

- the system will get the values.
- the system will ensure user has already selected an element.
- process the atom value to ensure it's a number. if not error massage will be shown.
- if its a number it will test if its greater than 0 or not. if not error massage will be shown.

4. **T4: get Mass Input**

Control: Functional, dynamic, automatic.

Initial State: Not Applicable.

Input:

input	response
Mass $\notin$ number	error massage
Mass ≤ 0	error massage
Mass > 0	accepted

Table 4: Input possible value's level

Output: error massage indicating that mass value is not a number or mass value ≤ 0.

Test Case Derivation: the expected value is a positive integer number.

How test will be performed:

- the system will get the value.
- process the value to ensure it's a number. if not error massage will be shown.
- if its a number it will test if its greater than 0 or not. if not error massage will be shown.

5. **T4: get reactant of known Mass**

Control: Functional, dynamic, automatic.

Initial State: Not Applicable.

Input: reactant number of known mass from drop down menu (1 or 2)

Output: error massage indicating that reactant number is not chosen.

Test Case Derivation: the expected value is to choose which reactant the known mass belong to. .

How test will be performed: the system will check if user choose the number of known reactant mass from drop down menu through GUI

widget. if not error massage will be shown indicating that he need to choose from the menu before calculating the mass.

5.1.2 Calculation Method Tests

Balancing Calculation Tests

1. **T5: Count total number of atoms for each element in each side**

   Control: Functional, dynamic, automatic.

   Initial State: N.A

   Input: FeCl_3 + MgO $\rightarrow$ Fe_2O_3 + MgCl_2

   output: Reactants= [Fe:1, Cl:3 , Mg:1, O:1] , products = [Fe:2, Cl:2 , Mg:1, O:3]

   Test Case Derivation: the expected value is total number of atoms for each element in each side of the reaction.

   How test will be performed:

   - the system will get chemical reaction elements and atoms value.
   - the system will calculate the total atom value for each element in each side of the reaction.
   - the system will save the total of each element in reactant dictionary and product dictionary

2. **T6: CheckBalanceTest**

   Control: Functional, dynamic, automatic.

   Initial State: N.A

   Input: Reactants= [Fe:1, Cl:3 , Mg:1, O:1] , products = [Fe:2, Cl:2 , Mg:1, O:3]

   output: "not balance!"

   Test Case Derivation: the expected value is balance when chemical reaction is balance and not balance otherwise.

   How test will be performed:

- the system will get reactant and product dictionaries for a reaction.
- the system will compare the total of each element.
- if total atoms value in reactant side = total atoms value in product side then the chemical reaction considered balance .
- "balance" will be displayed to the user using GUI widget.
- if not balance then balance form will be calculated and "not balance !" will be displayed to the user using GUI widget with the new form of the reaction.

3. **T7: BalanceTest**

   Control: Functional, dynamic, automatic.

   Initial State: N.A

   Input: Reactants= [Fe:1, Cl:3 , Mg:1, O:1] , products = [Fe:2, Cl:2 , Mg:1, O:3]

   output: Reactants1= [Fe: 2, Cl: 6, Mg: 3, O: 3] , products1 = [Fe: 2, O: 3, Mg: 3, Cl: 6]

   Test Case Derivation: the expected value is equal number of total atoms for each element in both dictionaries.

   How test will be performed:

   - the system will get reactant and product dictionaries for a reaction.
   - the system will compare the total of each element.
   - if total atoms value in reactant side = total atoms value in product side then the chemical reaction considered balance .
   - if not balance then balance form will be calculated and updated dictionaries will be generated.

4. **T8: Coefficients Test**

   Control: Functional, dynamic, automatic.

   Initial State: N.A

   Input:

- FeCl_3 + MgO $\rightarrow$ Fe_2O_3 + MgCl_2
- Reactants= [Fe:1, Cl:3 , Mg:1, O:1]
- products = [Fe:2, Cl:2 , Mg:1, O:3]
- Reactants1= [Fe: 2, Cl: 6, Mg: 3, O: 3]
- products1 = [Fe: 2, O: 3, Mg: 3, Cl: 6]

output: [Coefficient1:2/1= 2, Coefficient2: 3/1=3 , Coefficient3: 2/2=1 , Coefficient4: 3/1=3]

Test Case Derivation: the expected value is four Coefficients.

How test will be performed:

- the system will get all dictionaries generated (Reactants, Reactants1, products, products1) .
- the system will generate Coefficient1 which is equal to the total number of atoms of the first element of the reactant 1 after balancing(Reactants1) / total number of atoms of the element before balancing(Reactants).
- the system will generate Coefficient2 which is equal to the total number of atoms of the first element of the reactant 2 after balancing(Reactants1) / total number of atoms of the element before balancing(Reactants).
- the system will generate Coefficient3 which is equal to the total number of atoms of the first element of the product 1 after balancing(products1) / total number of atoms of the element before balancing(products).
- the system will generate Coefficient4 which is equal to the total number of atoms of the first element of the product 2 after balancing(products1) / total number of atoms of the element before balancing(products).
- the system will place Coefficient1 in front of reactant1 , Coefficient2 in front of reactant2 , Coefficient3 in front of product1 and Coefficient4 in front of product2 .

Mass Calculation Tests

1. **T9: atomic mass Test**

   Control: Functional, dynamic, automatic.

   Initial State: Not Applicable

   Input: name of element: "Fe"

   Output: atomic mass = 55.845

   Test Case Derivation: the expected value is atomic mass of an element.

   How test will be performed:

   - the system will get element name
   - the system will return the corresponding atomic mass
   - save the final result to be used in other function.

2. **T10: MolecularWeightCalculationTest**

   Control: Functional, dynamic, automatic.

   Initial State: Not Applicable

   Input: name of known reactant: "Fe2O3"

   Output: Molecular weight = 55.845*2 + 15.9994*3 = 159.6882 g/mol.

   Test Case Derivation: the expected value is a molecular weight which is equal to the sum of atomic mass multiplied by the total atom value for each element building up the reactant.

   How test will be performed:

   - the system will get the atomic mass for each element of the reactant.
   - multiply each atomic mass by the total atoms of the element
   - add the result of each element together.
   - save the final result to be used in other function.

3. **T11: Mole1CalculationTest**

   Control: Functional, dynamic, automatic.

   Initial State: Not Applicable

   Input: Mass = 2 g , molecular weight = 159.6882 g/mol

   Output: Mole = 2 / 159.6882 = 0.0125 mol.

   Test Case Derivation: the expected value is mole for reactant with known mass which is equal to given mass value divided by calculated molecular weight.

   How test will be performed:

   - the system will get mass and molecular weight for the reactant.
   - divide mass by molecular weight
   - save the final result to be used in other function.

4. **T12: MoleRatioCalculationTest**

   Control: Functional, dynamic, automatic.

   Initial State: Not Applicable

   Input: coefficient1 = 2 , coefficient2 = 3

   Output: coefficient2/coefficient1 = 3/2 = 1.5

   Test Case Derivation: the expected value is Mole Ratio which is equal to derived coefficient value of reactant with unknown mass divided by coefficient value of reactant with known mass.

   How test will be performed:

   - the system will get coefficient2 and coefficient1 derived from balancing the reaction..
   - divide coefficient2 by coefficient1
   - save the final result to be used in other function.

5. **T13: Mole2CalculationTest**

Control: Functional, dynamic, automatic.

Initial State: Not Applicable

Input: Mole1 = 0.0125 , MoleRatio = 1.5

Output: Mole1/MoleRatio = 0.0125/1.5 = 0.008

Test Case Derivation: the expected value is Mole2 (mole for reactant with unknown mass) which is equal to Mole1 value divided by MoleRatio.

How test will be performed:

- the system will get Mole1 and MoleRatio derived from other functions
- divide Mole1 by MoleRatio
- save the final result to be used in other function.

6. **T14: MassCalculationTest**

Control: Functional, dynamic, automatic.

Initial State: Not Applicable

Input: Mole2 = 0.0062 , Molecular weight = 12.0107

Output: Mole2 * Molecular weight = 0.008 * 12.0107 = 0.1 g.

Test Case Derivation: the expected value is final Mass which is equal to Mole2 value multiplied by reactant Molecular weight.

How test will be performed:

- the system will get Mole2 and Molecular weight derived from other functions
- multiply Mole2 by Molecular weight
- send the final result to GUI to be displayed.

5.1.3 Output Tests

Output constrains tests

1. **T15: ReactionOutputTest**

   Control: Functional, dynamic, automatic.

   Initial State: N.A

   Input: chemical reaction : "Fe_2O_3 + C $\rightarrow$ Fe + CO_2" two reactants and two products each with maximum two elements.

   Output: balance reaction : 2Fe_2O_3 + 3C $\rightarrow$ 4Fe + 3CO_2

   Test Case Derivation: the expected value is balance chemical reaction by adding the appropriate coefficients in front of each reactant and product.

   How test will be performed:

   - the system will get chemical reaction elements and atoms value.
   - if the entered reaction is not balance then the system will output the balanced reaction instead.
   - the balance result will be compared to (1) as parallel testing.

2. **T16: MassOutputTest**

   Control: Functional, dynamic, automatic.

   Initial State: Not Applicable

   Input:
   chemical reaction : "Fe_2O_3 + C $\rightarrow$ Fe + CO_2"
   mass of known reactant : 2
   name of known reactant: "Fe_2O_3"

   Output: C mass = 0.1 g.

   Test Case Derivation: the expected value is a positive integer number.

   How test will be performed: the system will calculate mass value using the calculation method and ensure it's greater than 0.

5.2 Tests for Nonfunctional Requirements

1. **T17: UsabilityTesting**

Type: Nonfunctional,Dynamic, Manual.

Initial State: Not Applicable.

Input/Condition: chemical reaction , mass of one reactant

Output/Result: Balance reaction , Second reactant mass.

How test will be performed: user need to complete small survey. see appendix A

2. **T18: Reliability**

Type: Functional, Dynamic, Manual, Static etc.

Initial State: Not Applicable.

Input: testing 25 unbalanced chemical reaction with given mass for one reactant.

Output: the percentage of correct answer. (aim for 100 % correct answer)

How test will be performed: the developer manually will test the overall system using 25 different examples of unbalanced chemical reaction consists of two reactant each consist of at most two elements with given mass of one of reactant, and get the percentage of correct answer. correct answer will include: (right balancing + correct mass value)

5.3 Traceability Between Test Cases and Modules

A trace between system tests and requirements is provided in Table 5.

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16
Input Module	X	X	X	X												
Atomic Mass Module									X							
Balancing Chemical Reaction Module					X	X	X	X								
Mass Calculation Module										X	X	X	X	X		
GUI Module															X	X

Table 5: Traceability Matrix Showing the Connections Between Modules and Test Cases

References

[1] http://www.endmemo.com/chem/balancer.php

6 Appendix

Survey to measure the Satisfaction of user on SMMP system:

	1	2	3	4	5
The ability to understand the propose of the system					
The ability to fill the input easily					
The system is easy to use					
The output is clear					
The GUI is visible and consistent					
Flexibility and efficiency of use					
I would like to use the system again					

- 1 is the lowest Satisfaction level and 5 is the highest.

Figure 1: user satisfaction survey.