System Verification and Validation Plan for ChemCode

Samuel Crawford

February 20, 2023

Contents

List of Tables

List of Figures

[Remove this section if it isn't needed —SS]

1 Revision History

Date	Version	Notes
Feb. 5, 2023	0.0	Create document and remove inapplicable content
Feb. 7-8, 2023	0.1.0	Add input tests
Feb. 8, 2023	0.1.1	Improve referencing of tests
	0.1.2	Add matrix conversion tests and improve input tests, including rationale, labelling, and chemical equations
Feb. 9, 2023	0.1.3	Add tests for trivial equation
	0.1.4	Add feasibility tests
Feb. 13, 2023	0.1.5	Clarify notion of matrices having the same solution after swapping rows and/or columns
Feb. 13-14, 2023	0.1.6	Add balancing and output tests
Feb. 14, 2023	0.1.7	Add accuracy test for balancing
Feb. 16, 2023	0.1.8	Move (reordered) "system tests" to unit test section (commented out), convert them to true system tests, and tweak balancing accuracy test appropriately
	0.1.9	Add static test for element support
Feb. 18, 2023	0.1.10	Improve formatting of existing tests, including adding references to requirements
Feb. 19, 2023	0.2.0	Add remaining tests for nonfunctional requirements (except for NFR4)
	0.2.1	Fill in Plan section
	0.3.0	Fill in General Information section
Feb. 20, 2023	0.3.1	Add references to requirements and external documents
	0.3.2	Finish Abbreviations and Acronyms section
	0.3.3	Fill in Unit Test Description section and remove its template content

2 Abbreviations and Acronyms

In addition to the ones from the SRS [1], the following abbreviations and acronyms are used throughout this document:

symbol	**description**
CAS	Computing and Software
SUS	System Usability Scale
T	Test

3 General Information

This document outlines the plan for verifying and validating ChemCode. It provides a brief overview of ChemCode, as well as the main objectives of this document and references to other relevant documentation, an overview of the verification and validation of various artifacts, a description of the system and unit tests, and any other relevant information.

3.1 Summary

The software that will be tested is ChemCode, a tool for automatically balancing chemical equations so that they may be useful [2].

3.2 Objectives

The primary objectives of this plan are to build confidence in the software's correctness and to ensure that this project meets the goals of both CAS 741 and Drasil.

3.3 Relevant Documentation

This plan outlines the process for verifying the SRS [1] and this VnV Plan in Sections 4.2 and 4.4, respectively; these are the only relevant pieces of documentation for this plan. (Note that design documentation would also be relevant, but since ChemCode is being built in Drasil which doesn't generate this documentation [3], it is out of the scope of this project.)

4 Plan

This section outlines the plan for verifying and validating ChemCode, which will be performed by a team. This plan includes verifying the SRS [1], design, VnV Plan, and implementation of ChemCode. It also includes the use of automated tools for testing and verification and external data for validation.

4.1 Verification and Validation Team

Each teammate has their role listed in the Table of Teammates and Their Roles with the following roles:

- **Author**: The person writing the documentation and Drasil implementation of ChemCode.
- **Project Supervisor**: The person in charge of the ChemCode project, including its implementation in Drasil.
- **Reviewer**: A person in charge of ensuring that the documentation and/or implementation of ChemCode meets the needs of the CAS 741 course (this may be limited to one artifact).
- **Validator**: A person in charge of ensuring that the documentation and/or implementation of ChemCode meets the goals of its implementation in Drasil.
- **Verifier**: A person in charge of ensuring that the implementation of ChemCode meets the requirements from its SRS.

Name	**Role**
Dr. Spencer Smith	Project Supervisor, Reviewer, Validator
Samuel Crawford	Author, Reviewer, Validator, Verifier
Jason Balaci	Reviewer, Validator
Deesha Patel	SRS Reviewer
Maryam Valian	VnV Plan Reviewer
Karen Wang	Drasil Implementation Reviewer
Drasil Team	Validators, Verifiers
Class of CAS 741	Reviewers

Table 1: Table of Teammates and Their Roles

4.2 SRS Verification Plan

The first round of review for the SRS will be done on its manual version. It will be reviewed first by Samuel Crawford during the writing process, then generally by the class of CAS 741, and then more rigorously by Jason Balaci, Deesha Patel, and Dr. Spencer Smith. After appropriate revisions and its implementation in Drasil, the generated version will then be reviewed by

Samuel Crawford, Dr. Spencer Smith and a subset of the Drasil team. Each review will check it against the general writing checklist [4], the SRS checklist [5] and the SRS rubric for CAS 741 [I couldn't find a link for this. —SC]; the only exception is that the Drasil team will only verify the SRS within the context of Drasil (so they will not check it against the CAS 741 rubric, for example).

4.3 Design Verification Plan

The design of ChemCode will be dictated by the code generation of Drasil [3]. As Drasil does not currently generate design documentation [3] and writing it is outside the scope of the ChemCode project, this verification will be conducted based on the generated code itself. Verifying ChemCode's design will be done by the Drasil team according to their process of design verification and is therefore out of the scope of this document.

4.4 Verification and Validation Plan Verification Plan

The VnV Plan will be reviewed first by Samuel Crawford during the writing process, then generally by the class of CAS 741, and then more rigorously by Jason Balaci, Maryam Valian, and Dr. Spencer Smith. After appropriate revisions, it will then be reviewed again by Samuel Crawford and Dr. Spencer Smith. Each review will check it against the general writing checklist [4], the VnV checklist [6], and the VnV rubric for CAS 741 [I couldn't find a link for this. —SC].

4.5 Implementation Verification Plan

The primary method of verifying the implementation of ChemCode will be performing the system and unit tests from Sections 5 and 6, respectively. Since the implementation will be generated by Drasil, a subset of the Drasil team will also inspect the code to ensure that it matches the expectations for generated code, especially focusing on the new code generation for solving linear systems. [Does the Drasil team already perform any formal static analysis? —SC] The implementation will also likely be reviewed by Drasil team members in the future, as related to future work on Drasil.

4.6 Automated Testing and Verification Tools

Since the initial implementation of ChemCode will be in Python, pytest will be used to automate testing and measure coverage where appropriate. Other frameworks for automated testing will be added if/when code for ChemCode is generated in more languages. The Drasil repository has continuous integration set up to ensure that hlint standards are adhered to and that the code compiles [double check —SC]. The Makefiles generated by Drasil [3] will also be used. Since the code for ChemCode will be generated by Drasil, the Drasil team may potentially use other tools as part of its verification (e.g., linters).

4.7 Software Validation Plan

There are two sets of goals for ChemCode: those for CAS 741 and those for Drasil. The verification of the SRS from Section 4.2 will help ensure that ChemCode satisfies both sets of goals.

[If there is any external data that can be used for validation, you should point to it here. —SS] [Would "external data" include a database of chemical reactions? —SC]

5 System Test Description

5.1 Tests for Functional Requirements

The tests in each section are given in order of increasing complexity/likelihood of the situation arising during use of ChemCode.

5.1.1 Element Support Testing

T1: **Test for Element Support**

Test Case Derivation: In order for the user to be able to work with any possible chemical reaction, they must be able to enter each of the currently known 118 elements.

How test will be performed: Static analysis will be performed against the list of elements from [7], ensuring that the symbol of every element can be used throughout the execution of ChemCode. This includes the

input stage (from R1), the conversion stage (from R2), and the output stage (from R4 and R6 from the SRS [1]).

5.1.2 Feasible Reaction Testing

The following tests are for equations of feasible chemical reactions and will be performed automatically. Providing the correct output for an inputted feasible chemical reaction satisfies R1, R2, R3, R5, and R6 from the SRS [1]. Note that the input/output format for each test is an abstract representation, as the specific format of each input/output is a design decision that is not made at this stage.

T2: **Test for Small Valid Equation**

Input: $O_2 \rightarrow O_3$ [8]

Output: $3\,O_2 \rightarrow 2\,O_3$ [8, p. 6]

Test Case Derivation: The inputted chemical equation is valid and trivial.

T3: **Test for Valid Equation**

Input: $C_2H_6 + O_2 \rightarrow CO_2 + H_2O$ [9]

Output: $2\,C_2H_6 + 7\,O_2 \rightarrow 4\,CO_2 + 6\,H_2O$ [9, p. 523]

Test Case Derivation: The inputted chemical equation is valid and relatively small, but larger than the trivial one from T2.

T4: **Test for Large Valid Equation**

Input: $KMnO_4 + HCl \rightarrow MnCl_2 + KCl + Cl_2 + H_2O$ [10]

Output: $2\,KMnO_4 + 16\,HCl \rightarrow 2\,MnCl_2 + 2\,KCl + 5\,Cl_2 + 8\,H_2O$ [10]

Test Case Derivation: The inputted chemical equation is valid and larger than the one from T3.

T5: **Test for Valid Equation with Nonstoichiometric Compound**

Input: $Fe_{0.95}O + O_2 \rightarrow Fe_2O_3$ [11]

Output: $80\,Fe_{0.95}O + 17\,O_2 \rightarrow 38\,Fe_2O_3$

Test Case Derivation: The inputted chemical equation contains a non-stoichiometric compound (i.e., one with a fractional subscript).

5.1.3 Infeasible Reaction Testing

The following tests are for equations of infeasible chemical reactions and will be performed automatically. Providing the correct output for an inputted infeasible chemical reaction satisfies R1, R2, R3, and R4 from the SRS [1]. Note that the input/output format for each test is an abstract representation, as the specific format of each input/output is a design decision that is not made at this stage.

T6: **Test for Valid Equation that is Infeasible due to Conservation of Mass Violation**

Input: $C_2H_6 \rightarrow CO_2 + H_2O$

Output: "This reaction is infeasible because O is present on one side of the equation but not the other, violating the Law of Conservation of Mass."

Test Case Derivation: The inputted chemical equation is infeasible since every element does not exist on both sides of the equation, which violates the Law of Conservation of Mass (TM1 from SRS [add link —SC]).

T7: **Test for Valid Equation that is Infeasible due to Overconstrained System**

Input: $K_4FeC_6N_6 + K_2S_2O_3 \rightarrow CO_2 + K_2SO_4 + NO_2 + FeS$ (modified from [9])

Output: "This reaction is infeasible because it is overconstrained."

Test Case Derivation: The inputted chemical equation is infeasible since each compound has more than one element, so changing any coefficient affects the number of some other element, causing a chain reaction that does not converge. There is no solution to this system other than the trivial solution ($\mathbf{0}$) [9].

5.2 Tests for Nonfunctional Requirements

There are no tests for maintainability because this is the responsibility of the Drasil team, as Drasil is responsible for generated the actual implementation of ChemCode.

5.2.1 Accuracy Testing

T8: **Test for Accuracy of Balancing**

Type: Dynamic, Automatic

Test Case Derivation: Chemical equations are only useful if they are balanced [2], so computed coefficients should be exact. Since these coefficients should be the smallest possible whole numbers [2], there is exactly one possible set of coefficients for each feasible chemical equation and exact equality between this expected output and the actual output can be checked by a computer (e.g., by using integer comparison).

How test will be performed: This verification will be done by performing the Tests for Functional Requirements using integer comparison and satisfies R5 and NFR1 from the SRS [1].

5.2.2 Understandability Testing

T9: **Test for Understandability of ChemCode**

Type: Dynamic, Manual

How test will be performed: New typical users will be asked to input a valid chemical reaction to be balanced by the system. At least USER_FRAC of users should be able to do so within FIRST_USE_TIME. This test satisfies NFR2 from the SRS [1].

5.2.3 Usability Testing

T10: **Test for Usability of ChemCode**

Type: Dynamic, Manual

How test will be performed: Typical users will be told how ChemCode works and then instructed to input the chemical equations from the Tests for Functional Requirements to be balanced. They will then be asked the questions from the System Usability Scale (SUS) [12] (see Usability Survey). The average score should be at least SUS_SCORE. This test satisfies NFR3 from the SRS [1].

5.2.4 Portability Testing

T11: **Test for Portability of ChemCode**

Type: Dynamic

Test Case Derivation: ChemCode may be used by a large variety of users who may have different systems.

How test will be performed: This verification will be performed by ensuring the correct programming language(s) is/are installed on both a Windows and macOS system and the Tests for Functional Requirements will be performed. This test satisfies NFR5 from the SRS [1].

5.2.5 Verifiability Testing

T12: **Test for Verifiability of ChemCode**

Type: Dynamic, Static

How test will be performed: All other tests from this plan will be performed successfully. This test satisfies NFR6 from the SRS [1].

5.3 Traceability Between Test Cases and Requirements

[Provide a table that shows which test cases are supporting which requirements. —SS]

6 Unit Test Description

As mentioned in the Design Verification Plan, design documentation will not be written manually or generated by Drasil. Therefore, any unit tests will be based on the generated code itself. Unit testing may be outside the scope of the ChemCode project [Is this true? —SC], but if not, this section will be filled in once the code for ChemCode is generated by Drasil.

References

[1] S. Crawford, "Software Requirements Specification for ChemCode: A program for solving chemistry problems." https://github.com/JacquesCarette/Drasil/blob/chemCode/People/Sam/ChemCode/SRS/SRS.pdf, 2023.

[2] L. Lund and Anoka-Ramsey Community College, *Introduction to Chemistry*. Cambridge and Coon Rapids, MN, USA: LibreTexts, Jan. 2023.

[3] J. Carette, S. Smith, J. Balaci, T.-Y. Wu, S. Crawford, D. Chen, D. Szymczak, B. MacLachlan, D. Scime, and M. Niazi, "Drasil," Feb. 2021.

[4] S. Smith, "Writing Checklist." https://github.com/smiths/capTemplate/blob/main/docs/Checklists/Writing-Checklist.pdf, 2023.

[5] S. Smith, "SRS and CA Checklist." https://github.com/smiths/capTemplate/blob/main/docs/Checklists/SRS-Checklist.pdf, 2022.

[6] S. Smith, "System Verification and Validation Plan Checklist." https://github.com/smiths/capTemplate/blob/main/docs/Checklists/VnV-Checklist.pdf, 2022.

[7] A. M. Helmenstine, "A List of the Elements of the Periodic Table," May 2020.

[8] D. W. Fahey and M. I. C. L. A. Hegglin, "Twenty Questions and Answers About the Ozone Layer: 2010 Update, Scientific Assessment of Ozone Depletion: 2010," tech. rep., World Meteorological Organization, Geneva, Switzerland, Mar. 2011.

[9] I. Hamid, "Balancing Chemical Equations by Systems of Linear Equations," *Applied Mathematics*, vol. 10, pp. 521–526, July 2019.

[10] S. Taylor, "Balancing Complex Chemical Equations with Chemical Equations Examples," June 2021.

[11] Doubtnut, "When non stoiciometric compound Fe0.95O is heated in presence of oxygen then it convents into Fe2O3. Which of the following statement is correct?."

[12] N. Thomas, "How To Use The System Usability Scale (SUS) To Evaluate The Usability Of Your Website," July 2015. Section: Terminology.

7 Appendix

7.1 Symbolic Parameters

Symbolic constants used in test cases are defined here for easy maintenance.

- FIRST_USE_TIME = 10 minutes
- SUS_SCORE = 75 (note that "the average System Usability Scale score is 68" [12])
- USER_FRAC = 80%

7.2 Usability Survey

This survey was taken from [12]. The user will be asked to answer the following questions on a scale of one to five, where one means "Strongly Disagree" and five means "Strongly Agree." The scores for all odd-numbered questions will be decremented by one, the even-numbered question scores will be subtracted from five, and the sum of these new values will be multiplied by 2.5 to yield the final score.

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.