Software Requirements Specification for Diagnose

Andrea Clemeno

December 2, 2020

Contents

1 Reference Material

This section records information for easy reference.

1.1 Table of Units

The unit system used throughout is SI (Système International d'Unités). In addition to the basic units, several derived units are also used. For each unit, Tab: ToU lists the symbol, a description and the SI name.

Symbol	Description	SI Name
copies	number of biological units	copies
mL	volume	millilitre
s	time	second

Table 1: Table of Units

I like mol better, but probably because that is what I am used to.

1.2 Table of Symbols

The symbols used in this document are summarized in Tab: ToS along with their units. Throughout the document, symbols in bold will represent vectors, and scalars otherwise. The symbols are listed in alphabetical order. For vector quantities, the units shown are for each component of the vector.

Symbol	Description	Units
N	Viral load	$\frac{\text{copies}}{\text{mL}}$
N_o	Initial viral load	$\frac{\text{copies}}{\text{mL}}$
N_p	Predicted viral load after 30 days	$\frac{\text{copies}}{\text{mL}}$
N_t	Viral load at time t	$\frac{\text{copies}}{\text{mL}}$
n	Number of virions	copies
r	Rate of change of the viral load	$\frac{\text{copies}}{\text{mLs}}$
t	Time	s
V	Volume	mL
λ	Elimination constant	s^{-1}

Table 2: Table of Symbols

1.3 Abbreviations and Acronyms

Abbreviation	Full Form
A	Assumption
DD	Data Definition
GD	General Definition
GS	Goal Statement
IM	Instance Model
PS	Physical System Description
R	Requirement
SRS	Software Requirements Specification
TM	Theoretical Model
Uncert.	Typical Uncertainty

Table 3: Abbreviations and Acronyms

2 Introduction

HIV-1 is a virus that attacks cells of the immune system needed to fight off diseases. The virus leads to an uncurable disease called AIDs. Therefore, it is useful to have a program to model these types of problems. The program documented here is called Diagnose.

The following section provides an overview of the Software Requirements Specification (SRS) for Diagnose. This section explains the purpose of this document, the scope of the requirements, the characteristics of the intended reader, and the organization of the document.

2.1 Scope of Requirements

the scope should be an abstract view of the assumptions

The scope of the requirements includes the analysis of HIV-1 concentration over time.

3 Specific System Description

This section first presents the problem description, which gives a high-level view of the problem to be solved. This is followed by the solution characteristics specification, which presents the assumptions, theories, and definitions that are used.

3.1 Problem Description

A system is needed to assess the risk before substantial immune destruction has occurred. The system will predict viral load at 30 days and the patient's progression.

3.1.1 Terminology and Definitions

This subsection provides a list of terms that are used in the subsequent sections and their meaning, with the purpose of reducing ambiguity and making it easier to correctly understand the requirements.

- Virus: Submicroscopic parasites that infect cells.
- Viral load: The concentration of HIV virus at a point in time.
- Infected cells: Cells that interact with the virus replicate into cells altered by the virus.
- Helper T cell: Cells of the immune system that neutralize infected cells.
- Elimination: Physical quantity undergoing a decline in amount.
- AIDs: Acquired Immunodeficiency Syndrome develops from an increase in HIV viral load to the extent where T cell count decreases to under 200 mol/L.
- Diagnosis: The determination of a patient's condition reached by a healthcare professional.
- Progression: The development towards a more advanced stage.

3.1.2 Physical System Description

The physical system of Diagnose, as shown in Fig:Virus, includes the following elements:

PS1: The HIV Virion.

PS2: The virus-infected cells.

PS3: The Helper T cell.

PS4: The Human Body.

3.1.3 Goal Statements

Given two HIV-1 viral load datum taken on consecutive days, the goal statements are:

detElimrate: Determine the elimination rate of the HIV Virus due to immune response.

predictVL30: Determine the viral load at 30 days.

3.2 Solution Characteristics Specification

The instance models that govern Diagnose are presented in Section: Instance Models. The information to understand the meaning of the instance models and their derivation is also presented, so that the instance models can be verified.

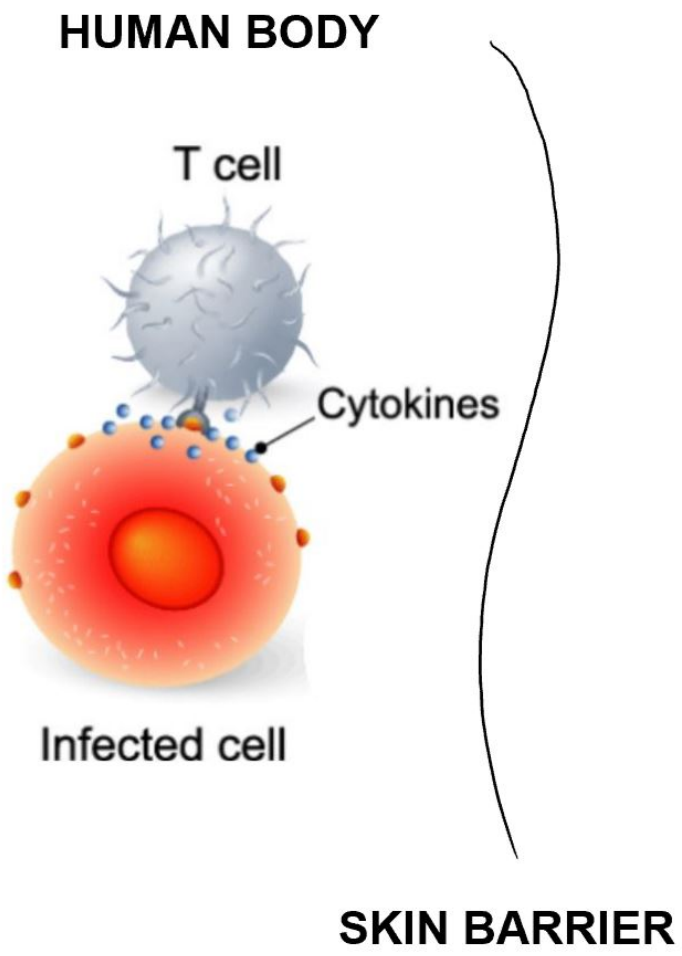

Figure 1: The physical system

3.2.1 Assumptions

This section simplifies the original problem and helps in developing the theoretical models by filling in the missing information for the physical system. The assumptions refine the scope by providing more detail.

initialInf: Initial infection of an HIV patient assumed. (RefBy: FR: Verify-Output, FR: Verify-Input-Values, IM: calofPredictedVL, and A: alwaysElim.)

constGrowth: The virions will invade uninfected cells at a constant rate.

constVolume: The dimensions of the location associated with the infection remains constant. (RefBy: DD: viralLoad.)

constConditions: Temperature of the location associated with the infection remains constant. (RefBy: A: neglectSick.)

allProductive: All infected cells are infect other cells productively. (RefBy: LC: More-Inputs.)

alwaysElim: In accordance with A: initialInf, after viremia peak, no significant upward trends occur.. (RefBy: FR: Verify-Output, FR: Verify-Input-Values, IM: calofPredictedVL, and IM: calofElimConst.)

neglectDrugs: The effect of antibiotic drugs or therapy on the elimination rate will be not be considered.. (RefBy: LC: More-Inputs.)

neglectSick: With reference to A: constConditions, the effect of other infections on the elimination rate will be not be considered.. (RefBy: LC: More-Inputs.)

proportional: The elimination of the virus is assumed to be proportional to the amount of viruses present. (RefBy: LC: Increase-time-frame and IM: calofElimConst.)

3.2.2 Theoretical Models

This section focuses on the general equations and laws that Diagnose is based on.

Refname	**TM:expElim**
Label	VLoad
Equation	$$r = \frac{dN}{dt} = -\lambda N_{\text{o}}$$
Description	r is the rate of change of the viral load ($\frac{\text{copies}}{\text{mLs}}$) t is the time (s) N is the viral load ($\frac{\text{copies}}{\text{mL}}$) λ is the elimination constant (s^{-1}) N_{o} is the initial viral load ($\frac{\text{copies}}{\text{mL}}$)
Source	[1]
RefBy	GD: vLoadt

3.2.3 General Definitions

This section collects the laws and equations that will be used to build the instance models.

Refname	**GD:vLoadt**
Label	VLoadt as a function of time for constant decay rate
Units	$\frac{\text{copies}}{\text{mL}}$
Equation	$N_\text{t} = N_\text{o} e^{-\lambda t}$
Description	N_t is the viral load at time t ($\frac{\text{copies}}{\text{mL}}$) N_o is the initial viral load ($\frac{\text{copies}}{\text{mL}}$) λ is the elimination constant (s^{-1}) t is the time (s)
Source	[3]
RefBy	IM: calofPredictedVL and IM: calofElimConst

Since you have a closed-form solution, you probably do not have to generate code for solving the ODE, which I wrote previously. You can just generate the explicit equation (like in the class BR example)

Detailed derivation of viral load at time t: Using the First-Order rate Law in TM: expElim, we have:

$$\frac{dN}{dt} = -\lambda N_\text{o}$$

Where N_t denotes the viral load at time t, N_o denotes the initial viral load and λ denotes the elimination constant . When rearranging for integration, we have:

$$\int_{N_\text{o}}^{N_\text{t}} 1 \, dN_\text{t} = -\int_0^t \lambda \, dt$$

Performing the integration, we have the required equation:

$$N_\text{t} = N_\text{o} e^{-\lambda t}$$

3.2.4 Data Definitions

This section collects and defines all the data needed to build the instance models.

Refname	**DD:viralLoad**
Label	Viral load
Symbol	N
Units	$\frac{\text{copies}}{\text{mL}}$
Equation	$N = \frac{n}{V}$
Description	N is the viral load ($\frac{\text{copies}}{\text{mL}}$) n is the number of virions (copies) V is the volume (mL)
Notes	The viral load describes the concentration of a virus within the body at a certain time. It assumes that the volume of blood is constant with respect to A: constVolume.
Source	–
RefBy	IM: calofPredictedVL and IM: calofElimConst

3.2.5 Instance Models

This section transforms the problem defined in Section: Problem Description into one which is expressed in mathematical terms. It uses concrete symbols defined in Section: Data Definitions to replace the abstract symbols in the models identified in Section: Theoretical Models and Section: General Definitions.

Refname	**IM:calofElimConst**
Label	Calculation of elimination rate
Input	N_t, t
Output	λ
Input Constraints	$0 < N_t < N_o$ $t > 0$
Output Constraints	$\lambda > 0$
Equation	$\lambda = \frac{\ln(N_o) - \ln(N_t)}{t}$
Description	λ is the elimination constant (s^{-1}) N_o is the initial viral load ($\frac{copies}{mL}$) N_t is the viral load at time t ($\frac{copies}{mL}$) t is the time (s)
Notes	The constraint $N_o > N_t > 0$ is required for the nature of the problem with respect to A: proportional. Due to the input constraint, the $\lambda > 0$ is established due to A: alwaysElim. Using this instance model, the goal of the software in GS: detElimrate can be achieved. In addition, this constraint is used to achieve a functional requirement seen in FR: Verify-Output.
Source	–
RefBy	IM: calofPredictedVL

Detailed derivation of elimination constant: Using the relationship for the viral load seen in DD: viralLoad with respect to time in GD: vLoadt, we have:

$$N_{\mathrm{t}} = N_{\mathrm{o}} e^{-\lambda t}$$

Where N_{t} denotes the viral load at time t, N_{o} denotes the initial viral load and λ denotes the elimination constant . When isolating for the elimination constant , we have:

$$\frac{N_{\mathrm{t}}}{N_{\mathrm{o}}} = e^{-\lambda t}$$

To isolate further, the natural logarithm is applied:

$$\ln\left(\frac{N_{\mathrm{t}}}{N_{\mathrm{o}}}\right) = -\lambda t$$

After using the logarithmic quotient property, we have the required equation:

$$\lambda = \frac{\ln\left(N_{\mathrm{o}}\right) - \ln\left(N_{\mathrm{t}}\right)}{t}$$

Refname	**IM:calofPredictedVL**
Label	Calculation of elimination rate
Input	N_o, t
Output	N_p
Input Constraints	$N_o > 0$ $t > 0$
Output Constraints	$0 < N_p < N_o$
Equation	$N_p = N_o e^{-\lambda t}$
Description	N_p is the predicted viral load after 30 days ($\frac{copies}{mL}$) N_o is the initial viral load ($\frac{copies}{mL}$) λ is the elimination constant (s^{-1}) t is the time (s)
Notes	The constraint $N_o > N_t > 0$ is applies when determining future values from the in initial infection with respect to A: initialInf as well as A: alwaysElim . Using this instance model, the goal of the software in GS: predictVL30 can be achieved. In addition, this constraint is used to achieve a functional requirement seen in FR: Verify-Output.
Source	–
RefBy	

Detailed derivation of predicted viral load after 30 days: Using the relationship for the viral load seen in DD: viralLoad with respect to time in GD: vLoadt, we have:

$$N_t = N_o e^{-\lambda t}$$

Where N_t denotes the viral load at time t, N_o denotes the initial viral load and λ denotes the elimination constant . When predicting the viral load after 30 days, the elimination constant found in IM: calofElimConst, is used instead of N_t therefore we have:

$$N_p = N_o e^{-\lambda t}$$

3.2.6 Data Constraints

Table:InDataConstraints shows the data constraints on the input variables. The column for physical constraints gives the physical limitations on the range of values that can be taken by the variable. The uncertainty column provides an estimate of the confidence with which the physical quantities can be measured. This information would be part of the input if one were performing an uncertainty quantification exercise. The constraints are conservative, to give the user of the model the flexibility to experiment with unusual situations. The column of typical values is intended to provide a feel for a common scenario.

Var	Physical Constraints	Typical Value	Uncert.
N_o	$N_o > 0$	$10.0 \cdot 10^6 \frac{\text{copies}}{\text{mL}}$	10%
N_t	$N_t > 0$	$5.0 \cdot 10^6 \frac{\text{copies}}{\text{mL}}$	10%

Table 4: Input Data Constraints

3.2.7 Properties of a Correct Solution

Table:OutDataConstraints shows the data constraints on the output variables. The column for physical constraints gives the physical limitations on the range of values that can be taken by the variable.

Var	Physical Constraints
λ	$\lambda > 0$
N_p	$N_p > 0$

Table 5: Output Data Constraints

You need to figure out exactly what your requirements are before you can move to the code gen. stage.

4 Requirements

This section provides the functional requirements, the tasks and behaviours that the software is expected to complete, and the non-functional requirements, the qualities that the software is expected to exhibit.

4.1 Functional Requirements

This section provides the functional requirements, the tasks and behaviours that the software is expected to complete.

Input-Values: Input the values from Table:ReqInputs.

Verify-Input-Values: The software will ensure that the inputs are not out of bounds and in accordance with the data constraints especially regarding the A: initialInf and furthermore, A: alwaysElim . If any inputs are out of bounds, an error message is displayed.

Calculate-Values: Software calculates the viral load at 30 days and the probability of progression to AIDs after 3 years.

How? What IMs?

How has this been monitored previously?

Verify-Output: The output values will be cross referenced with the result constraints, related to the assumptions: A: initialInf and A: alwaysElim.

what about CD4 (elimination rate)

Output-Values: Output related requirements .

Symbol	Description	Units
N_o	Initial viral load	$\frac{\text{copies}}{\text{mL}}$
N_t	Viral load at time t	$\frac{\text{copies}}{\text{mL}}$

Don't you need t?

Table 6: Required Inputs following FR: Input-Values

4.2 Non-Functional Requirements

This section provides the non-functional requirements, the qualities that the software is expected to exhibit.

Redundant - You don't need to say this

Correctness: The outputs have to adhere to the output properties in the output constraints.

Verifiable: The code is tested with complete verification and validation plan.

How you will do it, not the req.

Understandable: The code is modularized with complete module guide and module interface specification.

Reusable: The code is modularized.

Maintainable: The traceability between requirements, assumptions, theoretical models, general definitions, data definitions, instance models, likely changes, unlikely changes, and modules is completely recorded in traceability matrices in the SRS and module guide.

Portable: The code is able to be run in different environments.

5 Likely Changes

This section lists the likely changes to be made to the software.

ncrease-time-frame: The software may be expanded to cover a wide range of time frames which is possible due to A: proportional.

More-Inputs: The software may be expanded to include more inputs from the user to increase the accuracy of the output. This change may alter assumptions: A: allProductive , A: neglectSick and A: neglectDrugs.

More-Outputs: The software may be expanded to include more outputs like a suggestion for therapy.

6 Unlikely Changes

This section lists the unlikely changes to be made to the software.

ne-elimination-rate: The goal of determining the elimination rate of the virus will not likely change.

External-input: There will always be a source of input data external to the software.

g-input-constraints: The input constraints will not likely change.

7 Traceability Matrices and Graphs

The purpose of the traceability matrices is to provide easy references on what has to be additionally modified if a certain component is changed. Every time a component is changed, the items in the column of that component that are marked with an "X" should be modified as well. Table:TraceMatAvsA shows the dependencies of assumptions on the assumptions. Table:TraceMatAvsAll shows the dependencies of data definitions, theoretical models, general definitions, instance models, requirements, likely changes, and unlikely changes on the assumptions. Table:TraceMatRefvsRef shows the dependencies of data definitions, theoretical models, general definitions, and instance models with each other. Table:TraceMatAllvsR shows the dependencies of requirements, goal statements on the data definitions, theoretical models, general definitions, and instance models.

	A: initialInf	A: constGrowth	A: constVolume	A: constConditions	A
A: initialInf					
A: constGrowth					
A: constVolume					
A: constConditions					
A: allProductive					
A: alwaysElim	X				
A: neglectDrugs					
A: neglectSick				X	
A: proportional					

Table 7: Traceability Matrix Showing the Connect
each other.

	A: initialInf	A: constGrowth	A: constVolume	A: const
DD: viralLoad			X	
TM: expElim				
GD: vLoadt				
IM: calofElimConst				
IM: calofPredictedVL	X			
FR: Input-Values				
FR: Verify-Input-Values	X			
FR: Calculate-Values				
FR: Verify-Output	X			
FR: Output-Values				
NFR: Correctness				
NFR: Verifiable				
NFR: Understandable				
NFR: Reusable				
NFR: Maintainable				
NFR: Portable				
LC: Increase-time-frame				
LC: More-Inputs				
LC: More-Outputs				
UC: Determine-elimination-rate				
UC: External-input				
UC: Unchanging-input-constraints				

Table 8: Traceability Matrix Showing th
Items

	DD: viralLoad	TM: expElim	GD: vLoadt	IM: calofElimConst	IM
DD: viralLoad					
TM: expElim					
GD: vLoadt		X			
IM: calofElimConst	X		X		
IM: calofPredictedVL	X		X	X	

Table 9: Traceability Matrix Showing the Connections Between Items and Other Secti

	DD: viralLoad	TM: expElim	GD: vLoadt	IM: calofElimConst	I
GS: detElimrate					
GS: predictVL30					
FR: Input-Values					
FR: Verify-Input-Values					
FR: Calculate-Values					
FR: Verify-Output					
FR: Output-Values					
NFR: Correctness					
NFR: Verifiable					
NFR: Understandable					
NFR: Reusable					
NFR: Maintainable					
NFR: Portable					

8 Values of Auxiliary Constants

There are no auxiliary constants.

9 References

[1] Libretexts Contributors. *14.4: The Change of Concentration with Time (Integrated Rate Laws)*. https://chem.libretexts.org/Bookshelves/General_Chemistry/Map:_Chemistry_-_The_Central_Science_(Brown_et_al.)/14:_Chemical_Kinetics/14.4:_The_Change_of_Concentration_with_Time_(Integrated_Rate_Laws). Aug. 2020.

[2] Ulrike R. Fischer et al. *Benthic and Pelagic Viral Decay Experiments: a Model-Based Analysis and Its Applicability*. https://aem.asm.org/content/70/11/6706. June 2004.

[3] Russell K. Hobbie and Bradley J. Roth. *Exponential Growth and Decay*. https://link.springer.com/chapter/10.1007/978-0-387-49885-0_2. Jan. 1970.

[4] William C. Shiel Jr. MD. *Definition of T cell*. https://www.medicinenet.com/script/main/art.asp?articlekey=11300. Dec. 2018.

[5] Alan S. Perelson et al. *HIV-1 Dynamics in Vivo: Virion Clearance Rate, Infected Cell Life-Span, and Viral Generation Time*. https://science.sciencemag.org/content/271/5255/1582.full.pdf. June 1996.

[6] Eric W Weisstein. *Exponential Decay*. https://mathworld.wolfram.com/ExponentialDecay.html. June 1996.