

Software Requirements Specification for AortaGeomRecon

Jingyi Lin

March 14, 2023

Contents

1	Reference Material	iv
1.1	Table of Units	iv
1.2	Table of Symbols	iv
1.3	Abbreviations and Acronyms	v
2	Introduction	1
2.1	Purpose of Document	1
2.2	Scope of Requirements	1
2.3	Organization of Document	1
3	General System Description	1
3.1	System Context	1
3.2	User Characteristics	2
3.3	System Constraints	2
4	Specific System Description	2
4.1	Problem Description	3
4.1.1	Organ segmentation	3
4.1.2	Coordinate Systems	3
4.1.3	Physical System Description	6
4.1.4	Goal Statements	6
4.2	Solution Characteristics Specification	6
4.2.1	Assumptions	7
4.2.2	Theoretical Models	7
4.2.3	General Definitions	8
4.2.4	Data Definitions	9
4.2.5	Data Types	10
4.2.6	Instance Models	11
4.2.7	Input Data Constraints	11
4.2.8	Properties of a Correct Solution	12
5	Requirements	12
5.1	Functional Requirements	13
5.2	Nonfunctional Requirements	13
6	Likely Changes	14
7	Traceability Matrices and Graphs	14
8	Development Plan	15

Revision History

Date	Version	Notes
2023-02-12	1.0	Notes
2023-03-01	1.01	Modified system context image, coordinate systems, and goal statements.

1 Reference Material

This section records information for easy reference.

1.1 Table of Units

Throughout this document SI (Système International d’Unités) is employed as the unit system. In addition to the basic units, several derived units are used as described below. For each unit, the symbol is given followed by a description of the unit and the SI name.

1.2 Table of Symbols

The table that follows summarizes the symbols used in this document along with their units. The choice of symbols was made to be consistent with existing documentation for 3D Slicer program. The symbols are listed in alphabetical order.

symbol	type	description
MSE	\mathbb{R}	mean square error
MAE	\mathbb{R}	mean absolute error
$RMAE$	\mathbb{R}	root mean absolute error

Table 1: Table of Symbols

1.3 Abbreviations and Acronyms

symbol	description
A	Assumption
AortaGeomRecon	Aorta Geometry Reconstructor
DD	Data Definition
DICOM	Digital Imaging and Communications in Medicine
GD	General Definition
GS	Goal Statement
IM	Instance Model
LC	Likely Change
PS	Physical System Description
R	Requirement
SRS	Software Requirements Specification
SITK	Simple ITK
T	Theoretical Model

2 Introduction

This document provides an overview of the Software Requirements Specification (SRS) for the AortaGeomRecon. AortaGeomRecon provides a highly customizable aorta segmentation module, and an interactive user interface to apply the segmentation workflow.

2.1 Purpose of Document

The main purpose of this document is to provide sufficient information to understand what AortaGeomRecon module does. The goals and theoretical models used in the AortaGeomRecon segmentation module implementation are provided, with an emphasis on explicitly identifying assumptions and unambiguous definitions.

2.2 Scope of Requirements

2.3 Organization of Document

The organization of this document follows the template for an SRS for scientific computing software proposed by [Koothoor \(2013\)](#) and [Smith and Lai \(2005\)](#). The presentation follows the standard pattern of presenting goals, theories, definitions and assumptions. The goal statements are refined to the theoretical models, and theoretical models to the instance models. For readers that would like a more bottom-up approach, they can start reading the instance models in Section ?? and trace back to find any additional information they require.

3 General System Description

This section provides general information about the system. It identifies the interfaces between the system and its environment, describes the user characteristics and lists the system constraints.

3.1 System Context

Figure 1 shows the system context. A circle represents an external entity outside the software, the user in this case. A rectangle represents the software system itself. Arrows are used to show the data flow between the system and its environment.

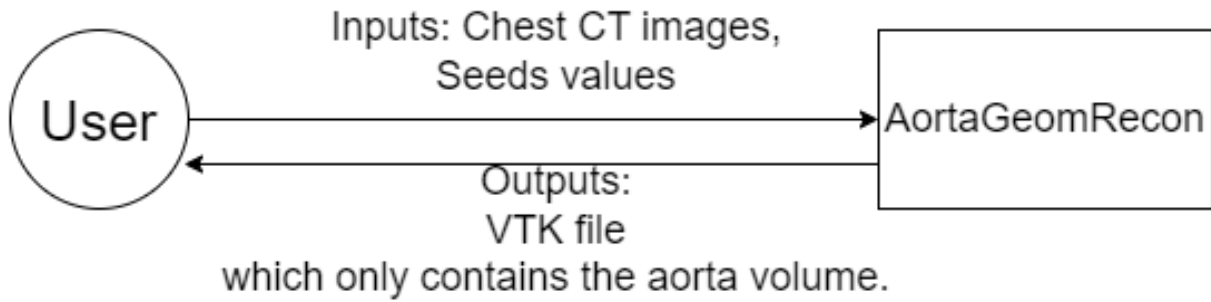


Figure 1: System Context

- User Responsibilities:
 - Provide the input data to the system
 - Ensure the input meets the necessary assumptions
 - Verify the result meets their requirements, otherwise repeat the process with a different seed values.
- AortaGeomRecon Responsibilities:
 - Provide DICOM data reader which can take a path to a folder containing DICOM files.
 - Provide crop functionality to easily select a region of interest.
 - Provide simple interactions to obtain and store the users' inputs. This includes a data probe to read voxel location which stored as a coordinate, and text inputs for real numbers.
 - Provide visualization on the result data.

3.2 User Characteristics

The end user of AortaGeomRecon have taking the university level anatomy introduction course, and is capable to point out the center of the descending aorta and the ascending aorta.

3.3 System Constraints

4 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, definitions and finally the instance models.

4.1 Problem Description

The main purpose of AortaGeomRecon is to semi-automatically segment a 3D aorta geometry from a chest CT scan.

4.1.1 Organ segmentation

The organ segmentation or the organ boundary is useful for orientation and identification of the regions of interests inside the organ during the diagnostic or treatment procedure. The aorta segmentation is important for aortic calcification quantification and to guide the segmentation of other central vessels.

4.1.2 Coordinate Systems

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.

While working with medical images, it is necessary to be familiar with the different coordinate systems of the medical literature and how data (voxels' orientation) is interpreted in different medical and non-medical software. Each coordinate system uses one or more numbers (coordinates) to uniquely determine the position of a point (in the medical context, we refer to each point as a voxel). The purpose of this section is to introduce some of the coordinate systems related to the medical imaging. There are different coordinate systems to represent data. A knowledge of the following coordinate systems is needed to work with the medical images.

Cartesian Coordinate System A Cartesian coordinate system is a coordinate system that specifies each point uniquely in a 2D plane by a pair of numerical coordinates or in a 3D space by three numerical coordinates. We assume a right-hand Cartesian coordinate system throughout this document.

World Coordinate System World Coordinate System (WCS) is a Cartesian coordinate system that describes the physical coordinates associated with a model such as an MRI scanner or a patient. While each model has its own coordinate system, without a universal coordinate system such as WCS, they cannot interact with each other. For model interaction to be possible, their coordinate systems must be transformed into the WCS. Figure 2 shows the WCS corresponding space and axes.

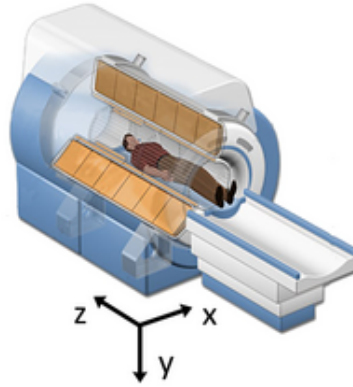


Figure 2: World Coordinate System Space and Axes [sli \(2014\)](#)

Anatomical Coordinate System Anatomical coordinate system, also known as patient coordinate system, is a right-handed 3D coordinate system that describes the standard anatomical position of a human using the following 3 orthogonal planes:

- Axial / Transverse plane: is a plane parallel to the ground that separates the body into head (superior) and tail (inferior) positions.
- Coronal / Frontal plane: is a plane perpendicular to the ground that divides the body into front (anterior) and back (posterior) positions.
- Sagittal / Median plane: is a plane that divides the body into right and left positions.

Figure 3 shows this coordinate system.

Medical applications follow an anatomical coordinate system to store voxels in sequences. Depending on how the data is stored, this coordinate system can be divided into different bases. The most common ones are:

- LPS Coordinate System:

The LPS coordinate system is used in DICOM images and by the ITK toolkit. In this system, voxels are ordered from left to right in a row, rows are ordered from posterior to anterior, and slices are stored from inferior to superior. LPS stands for Left-Posterior-Superior which indicates the directions that spatial axes are increasing.

- RAS Coordinate System:

The RAS coordinate system is the preferred basis for Neurological applications such as 3dfm+, and 3D Slicer. RAS stands for Right-Anterior-Superior is similar to LPS with the first two axes flipped.

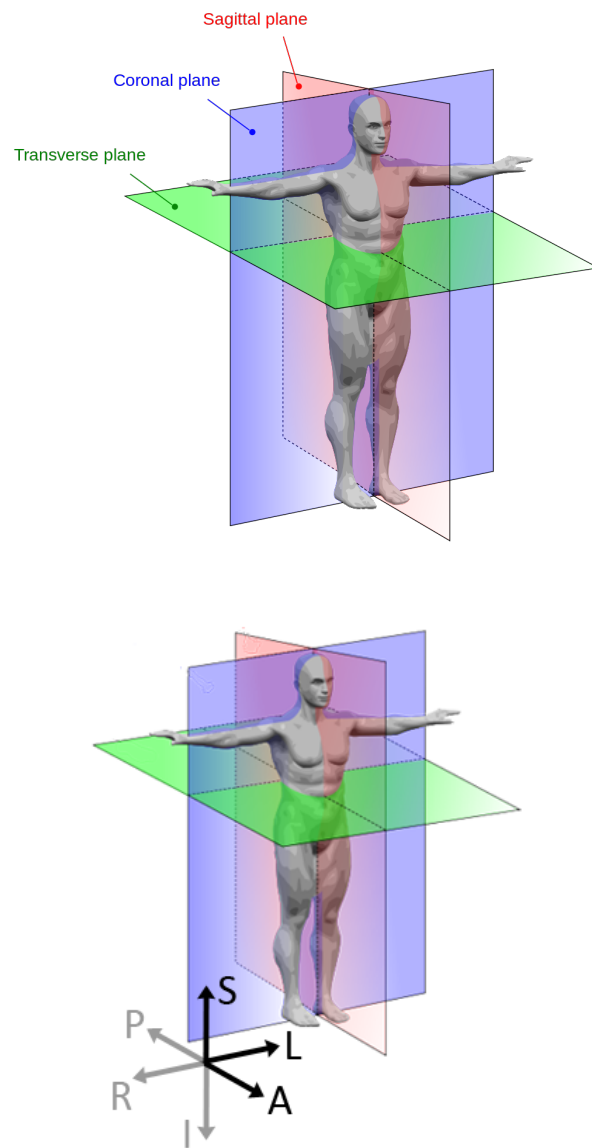


Figure 3: Anatomical Coordinate System Space and Axes [sli \(2014\)](#)

Image Coordinate System To specify locations in an image we need to know to which coordinate system it is referenced. Different software may use different orders as their index convention.

4.1.3 Physical System Description

We do not study the physical system for DICOM or how the data is actually generated.

4.1.4 Goal Statements

Given the DICOM image that includes patient's chest, the descending aorta center voxel coordinate, and the ascending aorta center voxel coordinate, the goal statements is:

GS1: Extract the 3 dimensional segmentation of the aorta.

4.2 Solution Characteristics Specification

[This section specifies the information in the solution domain of the system to be developed. This section is intended to express what is required in such a way that analysts and stakeholders get a clear picture, and the latter will accept it. The purpose of this section is to reduce the problem into one expressed in mathematical terms. Mathematical expertise is used to extract the essentials from the underlying physical description of the problem, and to collect and substantiate all physical data pertinent to the problem. —TPLT]

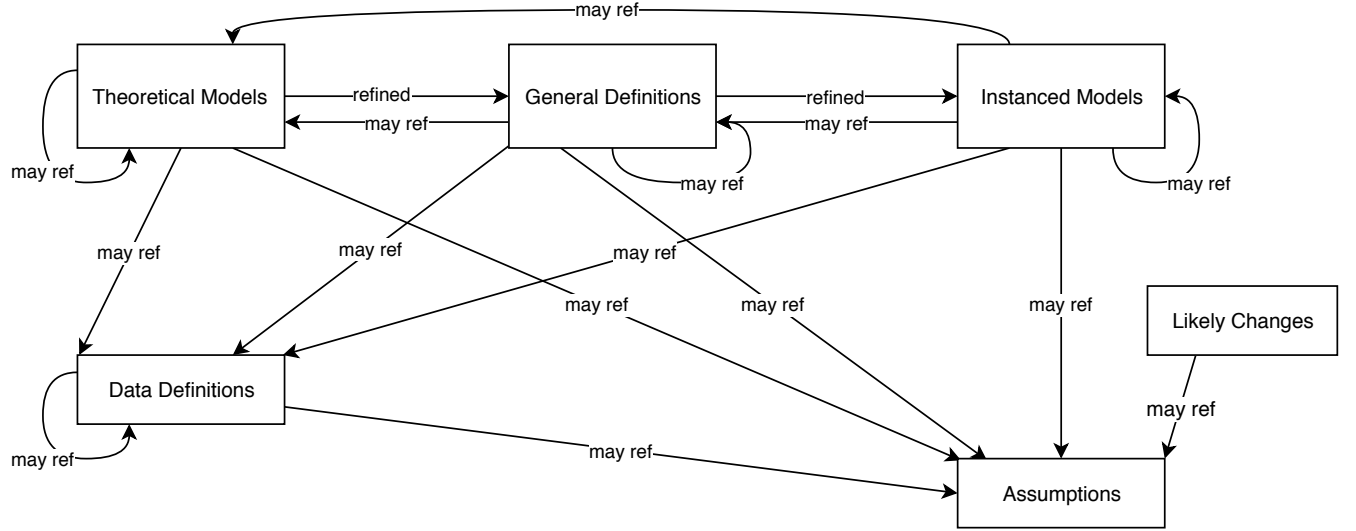
[This section presents the solution characteristics by successively refining models. It starts with the abstract/general Theoretical Models (TMs) and refines them to the concrete/specific Instance Models (IMs). If necessary there are intermediate refinements to General Definitions (GDs). All of these refinements can potentially use Assumptions (A) and Data Definitions (DD). TMs are refined to create new models, that are called GMs or IMs. DDs are not refined; they are just used. GDs and IMs are derived, or refined, from other models. DDs are not derived; they are just given. TMs are also just given, but they are refined, not used. If a potential DD includes a derivation, then that means it is refining other models, which would make it a GD or an IM. —TPLT]

[The above makes a distinction between “refined” and “used.” A model is refined to another model if it is changed by the refinement. When we change a general 3D equation to a 2D equation, we are making a refinement, by applying the assumption that the third dimension does not matter. If we use a definition, like the definition of density, we aren't refining, or changing that definition, we are just using it. —TPLT]

[The same information can be a TM in one problem and a DD in another. It is about how the information is used. In one problem the definition of acceleration can be a TM, in another it would be a DD. —TPLT]

[There is repetition between the information given in the different chunks (TM, GDs etc) with other information in the document. For instance, the meaning of the symbols, the units etc are repeated. This is so that the chunks can stand on their own when being read by a reviewer/user. It also facilitates reuse of the models in a different context. —TPLT]

[The relationships between the parts of the document are show in the following figure. In this diagram “may ref” has the same role as “uses” above. The figure adds “Likely Changes,” which are able to reference (use) Assumptions. —TPLT]



The instance models that govern AortaGeomRecon are presented in Subsection 4.2.6. The information to understand the meaning of the instance models and their derivation is also presented, so that the instance models can be verified.

4.2.1 Assumptions

This section simplifies the original problem and helps in developing the theoretical model by filling in the missing information for the physical system. The numbers given in the square brackets refer to the theoretical model [T], general definition [GD], data definition [DD], instance model [IM], or likely change [LC], in which the respective assumption is used.

A1: The 3D image provided by the user must contains a visually distinguishable aorta image [DD2].

4.2.2 Theoretical Models

[Theoretical models are sets of abstract mathematical equations or axioms for solving the problem described in Section “Physical System Description” (Section 4.1.3). Examples of theoretical models are physical laws, constitutive equations, relevant conversion factors, etc. —TPLT]

This section focuses on the general equations and laws that AortaGeomRecon is based on. [Modify the examples below for your problem, and add additional models as appropriate. —TPLT]

RefName: T:COE

Label: Conservation of thermal energy

Equation: $-\nabla \cdot \mathbf{q} + g = \rho C \frac{\partial T}{\partial t}$

Description: The above equation gives the conservation of energy for transient heat transfer in a material of specific heat capacity C ($\text{J kg}^{-1} \text{°C}^{-1}$) and density ρ (kg m^{-3}), where \mathbf{q} is the thermal flux vector (W m^{-2}), g is the volumetric heat generation (W m^{-3}), T is the temperature (°C), t is time (s), and ∇ is the gradient operator. For this equation to apply, other forms of energy, such as mechanical energy, are assumed to be negligible in the system (A??). In general, the material properties (ρ and C) depend on temperature.

Notes: None.

Source: http://www.efunda.com/formulae/heat_transfer/conduction/overview_cond.cfm

Ref. By: GD??

Preconditions for T:COE: None

Derivation for T:COE: Not Applicable

[“Ref. By” is used repeatedly with the different types of information. This stands for Referenced By. It means that the models, definitions and assumptions listed reference the current model, definition or assumption. This information is given for traceability. Ref. By provides a pointer in the opposite direction to what we commonly do. You still need to have a reference in the other direction pointing to the current model, definition or assumption. As an example, if T1 is referenced by G2, that means that G2 will explicitly include a reference to T1. —TPLT]

4.2.3 General Definitions

[General Definitions (GDs) are a refinement of one or more TMs, and/or of other GDs. The GDs are less abstract than the TMs. Generally the reduction in abstraction is possible through invoking (using/referencing) Assumptions. For instance, the TM could be Newton’s

Law of Cooling stated abstracting. The GD could take the general law and apply it to get a 1D equation. —TPLT]

This section collects the laws and equations that will be used in building the instance models.

[Some projects may not have any content for this section, but the section heading should be kept. —TPLT] [Modify the examples below for your problem, and add additional definitions as appropriate. —TPLT]

Number	GD1
Label	Newton's law of cooling
SI Units	W m^{-2}
Equation	$q(t) = h\Delta T(t)$
Description	<p>Newton's law of cooling describes convective cooling from a surface. The law is stated as: the rate of heat loss from a body is proportional to the difference in temperatures between the body and its surroundings.</p> <p>$q(t)$ is the thermal flux (W m^{-2}).</p> <p>h is the heat transfer coefficient, assumed independent of T (A??) ($\text{W m}^{-2} \text{ } ^\circ\text{C}^{-1}$).</p> <p>$\Delta T(t) = T(t) - T_{\text{env}}(t)$ is the time-dependent thermal gradient between the environment and the object ($^\circ\text{C}$).</p>
Source	Citation here
Ref. By	DD??, DD??

Detailed derivation of simplified rate of change of temperature

[This may be necessary when the necessary information does not fit in the description field. —TPLT] [Derivations are important for justifying a given GD. You want it to be clear where the equation came from. —TPLT]

4.2.4 Data Definitions

This section collects and defines all the data needed to build the instance models. The dimension of each quantity is also given.

Number	DD1
Label	Image/Slice
Symbol	<i>image</i> or <i>slice</i>
SI Units	
Equation	
Description	A visual representation of something that is represented using only two spatial dimensions with a sequence of arrays where real values represents the color or intensity.
Sources	
Ref. By	IM??, IM2

Number	DD2
Label	3D Image
Symbol	<i>3D_image</i>
SI Units	
Equation	
Description	A three dimensional image is a sequence of some images/slices.
Sources	
Ref. By	IM??

4.2.5 Data Types

This section collects and defines all the data types needed to document the models.

Type Name	Name for Type
Type Def	mathematical definition of the type
Description	description here
Sources	Citation here, if the type is borrowed from another source

4.2.6 Instance Models

This section transforms the problem defined in Section 4.1 into one which is expressed in mathematical terms. It uses concrete symbols defined in Section 4.2.4 to replace the abstract symbols in the models identified in Sections 4.2.2 and 4.2.3.

The goals GS1 are solved by finding IM1 and perform IM2 on the descending and ascending aorta.

Number	IM1
Label	Region of interest
Input	3D images
Output	3D images
Description	The regions of interest is a subset of the 3D image. This subset has the majority of the values which the users wants to read, process or extract.
Sources	
Ref. By	

Number	IM2
Label	Segmentation
Input	3D images
Output	3D images
Description	The process of extract a region of interest from the original 3D image. The extracted region of interest can be represented with high intensity pixel value. The rest of the image should have a lower intensity pixe valuel.
Sources	
Ref. By	

Derivation of ...

4.2.7 Input Data Constraints

Table 2 shows the data constraints on the input output variables. The column for physical constraints gives the physical limitations on the range of values that can be taken by the

variable. The column for software constraints restricts the range of inputs to reasonable values. The software constraints will be helpful in the design stage for picking suitable algorithms. 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. 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 specification parameters in Table 2 are listed in Table 3.

Table 2: Input Variables

Var	Physical Constraints	Software Constraints	Typical Value	Uncertainty
L	$L > 0$	$L_{\min} \leq L \leq L_{\max}$	1.5 m	10%

(*) [you might need to add some notes or clarifications —TPLT]

Table 3: Specification Parameter Values

Var	Value
L_{\min}	0.1 m

4.2.8 Properties of a Correct Solution

A correct solution must exhibit [fill in the details —TPLT]. [These properties are in addition to the stated requirements. There is no need to repeat the requirements here. These additional properties may not exist for every problem. Examples include conservation laws (like conservation of energy or mass) and known constraints on outputs, which are usually summarized in tabular form. A sample table is shown in Table 4 —TPLT]

[This section is not for test cases or techniques for verification and validation. Those topics will be addressed in the Verification and Validation plan. —TPLT]

5 Requirements

[The requirements refine the goal statement. They will make heavy use of references to the instance models. —TPLT]

Table 4: Output Variables

Var	Physical Constraints
T_W	$T_{\text{init}} \leq T_W \leq T_C$ (by A??)

This section provides the functional requirements, the business tasks that the software is expected to complete, and the nonfunctional requirements, the qualities that the software is expected to exhibit.

5.1 Functional Requirements

R1: Input the following functions, data and parameters:

symbol	description
X	CT Scans data in DICOM format (DD2))

R2: Output a region of interest that only contains aorta.

R3: Visualize the inputs or outputs data.

R4: [Verification related requirements. —TPLT]

R5: [Output related requirements. —TPLT]

[Every IM should map to at least one requirement, but not every requirement has to map to a corresponding IM. —TPLT]

5.2 Nonfunctional Requirements

[List your nonfunctional requirements. You may consider using a fit criterion to make them verifiable. —TPLT] [The goal is for the nonfunctional requirements to be unambiguous, abstract and verifiable. This isn't easy to show succinctly, so a good strategy may be to give a "high level" view of the requirement, but allow for the details to be covered in the Verification and Validation document. —TPLT] [An absolute requirement on a quality of the system is rarely needed. For instance, an accuracy of 0.0101 % is likely fine, even if the requirement is for 0.01 % accuracy. Therefore, the emphasis will often be more on describing how well the quality is achieved, through experimentation, and possibly theory, rather than meeting some bar that was defined a priori. —TPLT] [You do not need an entry for correctness in your NFRs. The purpose of the SRS is to record the requirements that

need to be satisfied for correctness. Any statement of correctness would just be redundant. Rather than discuss correctness, you can characterize how far away from the correct (true) solution you are allowed to be. This is discussed under accuracy. —TPLT]

- NFR1: **Accuracy** [Characterize the accuracy by giving the context/use for the software. Maybe something like, “The accuracy of the computed solutions should meet the level needed for <engineering or scientific application>. The level of accuracy achieved by AortaGeomRecon shall be described following the procedure given in Section X of the Verification and Validation Plan.” A link to the VnV plan would be a nice extra. —TPLT]
- NFR2: **Usability** [Characterize the usability by giving the context/use for the software. You should likely reference the user characteristics section. The level of usability achieved by the software shall be described following the procedure given in Section X of the Verification and Validation Plan. A link to the VnV plan would be a nice extra. —TPLT]
- NFR3: **Maintainability** [The effort required to make any of the likely changes listed for AortaGeomRecon should be less than FRACTION of the original development time. FRACTION is then a symbolic constant that can be defined at the end of the report. —TPLT]
- NFR4: **Portability** [This NFR is easier to write than the others. The systems that AortaGeomRecon should run on should be listed here. When possible the specific versions of the potential operating environments should be given. To make the NFR verifiable a statement could be made that the tests from a given section of the VnV plan can be successfully run on all of the possible operating environments. —TPLT]
- Other NFRs that might be discussed include verifiability, understandability and reusability.

6 Likely Changes

- LC1: IM2 There are various segmentation algorithms, each has a different procedure and inputs.

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” may have to be modified as well. Table 5 shows the dependencies of theoretical models, general definitions, data definitions, and instance models with each other. Table 6 shows the dependencies

of instance models, requirements, and data constraints on each other. Table 7 shows the dependencies of theoretical models, general definitions, data definitions, instance models, and likely changes on the assumptions.

[You will have to modify these tables for your problem. —TPLT]

[The traceability matrix is not generally symmetric. If GD1 uses A1, that means that GD1’s derivation or presentation requires invocation of A1. A1 does not use GD1. A1 is “used by” GD1. —TPLT]

[The traceability matrix is challenging to maintain manually. Please do your best. In the future tools (like Drasil) will make this much easier. —TPLT]

	T??	T??	T??	GD1	GD??	DD??	DD??	DD??	DD??	IM??	IM??	IM??	IM??
T??													
T??			X										
T??													
GD1													
GD??	X												
DD??				X									
DD??				X									
DD??													
DD??								X					
IM??					X	X	X				X		
IM??					X		X		X	X			X
IM??		X											
IM??		X	X				X	X	X		X		

Table 5: Traceability Matrix Showing the Connections Between Items of Different Sections

The purpose of the traceability graphs is also to provide easy references on what has to be additionally modified if a certain component is changed. The arrows in the graphs represent dependencies. The component at the tail of an arrow is depended on by the component at the head of that arrow. Therefore, if a component is changed, the components that it points to should also be changed. Figure ?? shows the dependencies of theoretical models, general definitions, data definitions, instance models, likely changes, and assumptions on each other. Figure ?? shows the dependencies of instance models, requirements, and data constraints on each other.

8 Development Plan

[This section is optional. It is used to explain the plan for developing the software. In particular, this section gives a list of the order in which the requirements will be implemented.

	IM??	IM??	IM??	IM??	4.2.7	R??	R??
IM??		X				X	X
IM??	X			X		X	X
IM??						X	X
IM??		X				X	X
R??							
R??						X	
R??					X		
R??	X	X				X	X
R??	X						
R??		X					
R??			X				
R??				X			
R4			X	X			
R??		X					
R??		X					

Table 6: Traceability Matrix Showing the Connections Between Requirements and Instance Models

	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??
T??	X																		
T??																			
T??																			
GD1		X																	
GD??			X	X	X	X													
DD??							X	X	X										
DD??			X	X						X									
DD??																			
DD??																			
IM??											X	X		X	X	X			X
IM??												X	X			X	X	X	
IM??														X					X
IM??													X					X	
LC??				X															
LC??								X											
LC??									X										
LC??											X								
LC??												X							
LC??															X				

Table 7: Traceability Matrix Showing the Connections Between Assumptions and Other Items

In the context of a course this is where you can indicate which requirements will be implemented as part of the course, and which will be “faked” as future work. This section can be organized as a prioritized list of requirements, or it could should the requirements that will be implemented for “phase 1”, “phase 2”, etc. —TPLT]

9 Values of Auxiliary Constants

[Show the values of the symbolic parameters introduced in the report. —TPLT]

[The definition of the requirements will likely call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance. —TPLT]

[The value of FRACTION, for the Maintainability NFR would be given here. —TPLT]

References

Coordinate systems, June 2014. URL https://www.slicer.org/wiki/Coordinate_systems.

Nirmitha Koothoor. A document drive approach to certifying scientific computing software. Master's thesis, McMaster University, Hamilton, Ontario, Canada, 2013. URL <http://hdl.handle.net/11375/13266>.

W. Spencer Smith and Lei Lai. A new requirements template for scientific computing. In J. Ralyté, P. Ågerfalk, and N. Kraiem, editors, *Proceedings of the First International Workshop on Situational Requirements Engineering Processes – Methods, Techniques and Tools to Support Situation-Specific Requirements Engineering Processes, SREP'05*, pages 107–121, Paris, France, 2005. In conjunction with 13th IEEE International Requirements Engineering Conference.

[The following is not part of the template, just some things to consider when filing in the template. —TPLT]

[Grammar, flow and L^AT_EX advice:

- For Mac users *.DS_Store should be in .gitignore
- L^AT_EX and formatting rules
 - Variables are italic, everything else not, includes subscripts ([link to document](#))
 - * [Conventions](#)
 - * Watch out for implied multiplication
 - Use BibTeX
 - Use cross-referencing
- Grammar and writing rules
 - Acronyms expanded on first usage (not just in table of acronyms)
 - “In order to” should be “to”

—TPLT]

[Advice on using the template:

- Difference between physical and software constraints
- Properties of a correct solution means *additional* properties, not a restating of the requirements (may be “not applicable” for your problem). If you have a table of output constraints, then these are properties of a correct solution.
- Assumptions have to be invoked somewhere
- “Referenced by” implies that there is an explicit reference
- Think of traceability matrix, list of assumption invocations and list of reference by fields as automatically generatable
- If you say the format of the output (plot, table etc), then your requirement could be more abstract

—TPLT]