Software Requirements Specification for Sun Catcher

Sharon (Yu-Shiuan) Wu

November 25, 2019

Contents

Revision History

Date	Version	Notes
2019/10./16	1.0	First version of SRS update to GitHub
Date 2	1.1	Notes

1 Reference Material

This section records information for easy reference.

1.1 Table of Units

Throughout this document SI (Système International d'Unités) is employedas 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.

symbol	unit	SI
$^\circ$	angle	degree
$'$	angle	-
k	mass	kilograms
m	length	metre
W	power	Watt ($\mathrm{W} = \mathrm{J\,s^{-1}}$)

[defining a date type is a great idea, but dates are not units —SS] [I have deleted the "date type" from the units table. Moreover redefine the Symbols of date more explicitly, so I can avoid placing months, days, and years in the same symbol. —Author] [N and S are not units —SS] [I will treat N and S direction as a positive and negative symbol. —Author]

[Only include the units that your SRS actually uses. —TPLT]

[Derived units, like newtons, pascal, etc, should show their derivation (the units they are derived from) if their constituent units are in the table of units (that is, if the units they are derived from are used in the document). For instance, the derivation of pascals as $\mathrm{Pa} = \mathrm{N\,m^{-2}}$ is shown if newtons and m are both in the table. The derivations of newtons would not be shown if kg and s are not both in the table. —TPLT]

[The symbol for units named after people use capital letters, but the name of the unit itself uses lower case. For instance, pascals use the symbol Pa, watts use the symbol W, teslas use the symbol T, newtons use the symbol N, etc. The one exception to this is degree Celsius. Details on writing metric units can be found on the NIST web-page. —TPLT]

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 the heat transfer literature and with existing documentation for solar water heating systems. The symbols are listed in alphabetical order.

symbol	unit	description
$\theta_{S_{\mathrm{date}}}$	$^\circ$	zenith angle of sun in the date of a sequence of day

θ_T	$^\circ$	the tilt angle for adjusting the solar panel
Φ_P	$(^\circ, ')$	the latitude of the solar panel [the units are degrees. N and S are not units. I can see that this could be confusing. Defining a data type will help you. I'll include some details below. —SS] [Yes, I it fixed by treating the N and S as + and - —Author]
δ_{date}	$^\circ$	the daily declination of the vertical noon sun in the date duration
I_S	$\frac{\mathrm{kW}}{\mathrm{m}^2}$	the intensity of the sun measured by the satellites
$I_{S_{\mathrm{daily}}}$	$\frac{\mathrm{kW}}{\mathrm{m}^2}$	the daily solar intensity of the noon in the date duration($date$).
$I_{S_{\mathrm{total}}}$	$\frac{\mathrm{kW}}{\mathrm{m}^2}$	the total solar intensity of the noon in the date duration($date$).
$I_{S_{\mathrm{max}}}$	$\frac{\mathrm{kW}}{\mathrm{m}^2}$	the hightest solar intensity of the noon in the date duration($date$).
P_E	kW	the estimated solar panel output energy
$P_{A_{\mathrm{h}}}$	m^2	the height of the solar panel
$P_{A_{\mathrm{w}}}$	m^2	the width of the solar panel
P_{r}	%	solar panel yield or efficiency
PR	%	performance ratio, coefficient for losses
$year_{\mathrm{Start}}$	-	the year of the calcuation's starting date
$month_{\mathrm{Start}}$	-	the month of the calcuation's starting date
day_{Start}	-	the day of the calcuation's starting date
$year_{\mathrm{End}}$	-	the year of the calcuation's ending date
$month_{\mathrm{End}}$	-	the month of the calcuation's ending date

[Use your problems actual symbols. The si package is a good idea to use for units. —TPLT]

1.3 Abbreviations and Acronyms

symbol	**description**
A	Assumption
DD	Data Definition
GD	General Definition
GS	Goal Statement
IM	Instance Model
LC	Likely Change
PS	Physical System Description
R	Requirement
SRS	Software Requirements Specification
SC	Sun Catcher
T	Theoretical Model

[Add any other abbreviations or acronyms that you add —TPLT]

[This SRS template is based on Smith and Lai [1], Smith et al. [2]. It will get you started. You should not modify the section headings, without first discussing the change with the course instructor. Modification means you are not following the template, which loses some of the advantage of a template, especially standardization. Although the bits shown below do not include type information, you may need to add this information for your problem. If you are unsure, please can ask the instructor. —TPLT]

[Feel free to change the appearance of the report by modifying the LaTeX commands. —TPLT]

[This template document assumes that a single program is being documented. If you are documenting a family of models, you should start with a commonality analysis. A separate template is provided for this. For program families you should look at [3, 4]. Single family member programs are often programs based on a single physical model. General purpose tools are usually documented as a family. Families of physical models also come up. —TPLT]

[The SRS is not generally written, or read, sequentially. The SRS is a reference document. It is generally read in an ad hoc order, as the need arises. For writing an SRS, and for reading one for the first time, the suggested order of sections is:

- Goal Statement
- Instance Models
- Requirements
- Introduction
- Specific System Description

—TPLT]

[Guiding principles for the SRS document:

- Do not repeat the same information at the same abstraction level. If information is repeated, the repetition should be at a different abstraction level. For instance, there will be overlap between the scope section and the assumptions, but the scope section will not go into as much detail as the assumptions section.

—TPLT]

[The template description comments should be disabled before submitting this document for grading. —TPLT]

[You can borrow any wording from the text given in the template. It is part of the template, and not considered an instance of academic integrity. Of course, you need to cite the source of the template. —TPLT]

[When the documentation is done, it should be possible to trace back to the source of every piece of information. Some information will come from external sources, like terminology. Other information will be derived, like General Definitions. —TPLT]

[An SRS document should have the following qualities: unambiguous, consistent, complete, validatable, abstract and traceable. —TPLT]

[The overall goal of the SRS is that someone that meets the Characteristics of the Intended Reader (Section 2.3) can learn, understand and verify the captured domain knowledge. They should not have to trust the authors of the SRS on any statements. They should be able to independently verify/derive every statement made. —TPLT]

2 Introduction

Due to the increasing concepts of creating an earth-friendly environment, the kits using renewable energy becomes more popular in the market. Solar energy is the most common type of renewable resource for a home. However, it is an expensive technology, and its cell efficiency is restricted by seasons. Therefore, Sun Catcheris created for home users to gain optimum energyfrom daily sunlight.

The following sections provide an overview of the Software Requirements Specification (SRS) for Sun Catcher. [remember to check with the writingchecklists! —SS][OK —Author] This section explains the purpose and the organization of the document, the scope of the requirements, and the characteristics of the intendedreader. [You should put use 80 character wide columns of text in your texfile. It is better for diffs, and easier to read on different editors. —SS][OK —Author]

[The introduction section is written to introduce the problem. It starts general and focuses on the problem domain. The general advice is to start witha paragraph or two that describes the problem, followed by a "roadmap" paragraph. A roadmap orients the reader by telling them what sub-sections to expect in the Introduction section. —TPLT]

[Rather than iffalse, you could have just set comments to false in the Comments.tex file. Although as it is set up right now, you would get all of the comments on, the wss comments and the plt comments. Therefore, I like your solution better. Another option would be to delete the plt comments. —SS]

2.1 Purpose of Document

The purpose of this document is to record the correct requirements of SC. The goal statement provided readers a consistent idea of what problem is solved. Thetheoretical models and the instance models, which state the mathematical terms supporting the theoretical models, are explained unambiguously for readers to reuse and verify the software. In the section of System Constraints, its contents will stay abstract because the content should only say what problem is being solved, but not how to solve it.

This document will be used as a starting point for subsequent development phases, including writing the design specification and the software verificationand validation plan. The design document will show how the requirements are to be realized, includingdecisions on the numerical algorithms and programming environment. The verification and validation plan will show the steps that will be used to increase confidence in the software documentation and the implementation.

[This section summarizes the purpose of the SRS document. It does not focus on the problem itself. The problem is described in the "Problem Description" section (Section 4.1). The purpose is for the document inthe context of the project itself, not in the context of the CAS 741 course. Although the "purpose" of the document is to get a grade in 741, you should not mention this. Instead, "fake it" as if this is a real project. The purpose section will be similar between projects. The purpose of the document is the purpose of the SRS, including communication, planning for the design stage, etc. —TPLT]

2.2 Scope of Requirements

The scope of the requirements includes stability analysis of a two-dimensional (2D) solar panel and the sun as the solar resource. The solar panel is assumed to place in a location near sea level, and the sky view above is unobstructed, with no trees, hills, clouds, dust, or haze ever blocking the sun. Based on the assumption, Sun Catcheris designed to calculate the ideal energy output. However, the resulting angle might not be ideal for the actual cases because of the different environments and the variable panels. The uses should adjust a different of a few angle to match their actual conditions. [Clarify that for the purpose of the calculations you are making these assumptions. The actual solar panel will never reach the ideal energy output given in the calculations by Sun Catcher. —SS][Yes, I should make the assumption more explicitly here —Author] [I tried to use Sun Catcher, but you haven't redefined it to SC. You should do tha that in the Common.tex file. —SS][Fixed, please try to search it again:) —Author]

[Modelling the real world requires simplification. The full complexity of the actual physics, chemistry, biology is too much for existing models, and for existing computational solution techniques. Rather than say what is in the scope, it is usually easier to say what is not. You can think of it as the scope is initially everything, and then it is constrained to create the actual scope. For instance, the problem can be restricted to 2 dimensions, or it can ignore the effect of temperature (or pressure) on the material properties, etc. —TPLT]

[The scope section is related to the assumptions section (Section 4.2.1). However, the scope and the assumptions are not at the same level of abstraction. The scope is at a high level. The focus is on the "big picture" assumptions. The assumptions section lists, and describes, all of the assumptions. —TPLT]

2.3 Characteristics of Intended Reader

The end-users of SC is expecting to understand the Calculation method in Grade 11 such as angle addition and subtraction theorems and understand the celestial mechanics regarding the formation of the latitude and the earth's tilt angle in the four seasons.

[Usually these characteristics are given in terms of education level, and possibly education specialization. You also want to mention that the reader should understand the longitude and latitude coordinate system. —SS]

[This section summarizes the skills and knowledge of the readers of the SRS. It does NOT have the same purpose as the "User Characteristics" section (Section 3.2). The intended readers are the people that will read, review and maintain the SRS. They are the people that will conceivably design the software that is intended to meet the requirements. The user, on the other hand, is the person that uses the software that is built. They may never read this SRS document. Of course, the same person could be a "user" and an "intended reader." —TPLT]

[The intended reader characteristics should be written as unambiguously and as specifically as possible. Rather than say, the user should have an understanding of physics, say what kind of physics and at what level. For instance, is high school physics adequate, or should the reader have had a graduate course on advanced quantum mechanics? —TPLT]

2.4 Organization of Document

The organization of this document follows the template for an SRS for scientificcomputing software proposed by [5] as well as [1]. The presentation follows the standard pattern of presenting goals, theories, definitions, and assumptions. For readers that would like a more bottom up approach, they can start reading the instance models in Section: Instance Models) and trace back to find any additional information they require. The goal statements (Section: Goal Statements) are refined to the theoretical models and the theoretical models (Section: Theoretical Models) to the instance models (Section: Instance Models). The instance models provide the set ofalgebraic equations that must be solved.

[This section provides a roadmap of the SRS document. It will help the reader orient themselves. It will provide direction that will help them select which sections they want to read, and in what order. This section will besimilar between project. —TPLT]

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. [This text can likely be borrowed verbatim. —TPLT]

[The purpose of this section is to provide general information about the system so the specific requirements in the next section will be easier to understand. The general system description section is designed to be changeable independent of changes to the functional requirements documented in the specific system description. The general system description provides a context for a family of related models. The general description can stay the same, while specific details are changed between family members. —TPLT]

3.1 System Context

System Context shows the design pattern of this program from users to the system then the output. The circle represents the users of this software. The rectangle represents the software system: Sun Catcher(SC). The arrow represents the inputsthat drive the software and the output expecting from the software system.

[Your system context will include a figure that shows the abstract view of the software. Often in a scientific context, the program can be viewed abstractly following the design pattern of Inputs → Calculations → Outputs. The system context will therefore often follow this pattern. The user provides inputs, the system does the calculations, and then provides the outputs to the user. The figure should not show all of the inputs, just an abstract view of the main categories of inputs (like material properties, geometry, etc.). Likewise, the outputs should be presented from an abstract point of view. In some cases the diagram will show other external entities, besides the user. For instance, when the software product is a library, the user will be another software program, not an actual end user. —TPLT]

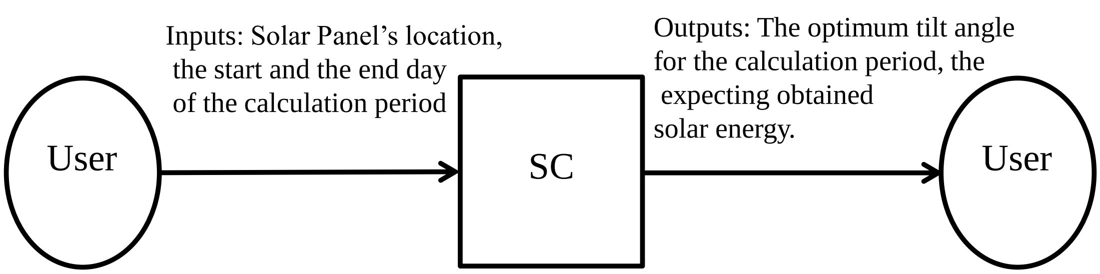

Figure 1: System Context

[For each of the entities in the system context diagram its responsibilitiesshould be listed. Whenever possible the system should check for data quality, but for some cases the user will need to assume that responsibility. —TPLT]

- User Responsibilities:

- Provide the input data related to the solar panel.
- Ensure the input is formatted correctly [I think you can just say "Ensure the input is formatted correctly." —SS] [Ok, I indeed can make it more abstract —Author]
- Ensure the input value are respond to the input constraint [The responsibilities are for using the software, not for using a solar panel. I don't think you need this responsibility. —SS][Delete the origin content. Yes, this condition should just be placed in the assumption not here. —Author]

- SC Responsibilities:
  - Detect data type mismatch, such as a string of characters instead of a floating point number.
  - Confirm [spellcheck! —SS][Yes —Author] the inputs quality to satisfy the required physical and software constraints.
  - Output the calculated result that satisfied the Goal Statements [You can write this more abstractly. The specifics of the output will be given in your requirements. —SS] [I changed the statement basing on the advice. —Author]

3.2 User Characteristics

This section summarized the expectation of the readers' knowledge and skills forunderstanding this SRS. Readers should have a general knowledge of how a solar panel works and know the common factors that affect energy absorption.

[I don't think Level 1 Calculation and Level 1 celestial mechanics are standard terms. Are they? Also, your user characteristics are more involved than your intended reader characteristics. Did you get these two sections backwards? —SS]

[This section summarizes the knowledge/skills expected of the user. Measuring usability, which is often a required non-function requirement, requires knowledge of a typical user. As mentioned above, the user is a different role from the "intended reader," as given in Section 2.3. As in Section 2.3, the user characteristics should be specific an unambiguous. For instance, "The end user of SC should have an understanding of undergraduate Level 1 Calculus and Physics." —TPLT]

3.3 System Constraints

There are no system constraints.

[System constraints differ from other type of requirements because they limit the developers' options in the system design and they identify how the eventual system must fit into the world. This is the only place in the SRS where design decisions can be specified. That is, the quality requirement for abstraction is relaxed here. However, system constraints should only be included if they are truly required. In the context of CAS 741, you often will may not have any system constraints. —TPLT]

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 finallythe instance models. [Add any project specific details that are relevant for the section overview. —TPLT]

4.1 Problem Description

Sun Catcheris intended to solve the unpredictable energy efficiency of solar panels. Due to the tilt angle of the earth when it rotates by axis, the latitude of the sun will consistently move. With the fixed location of the solarpanel, it is unlikely to get the direct the direct sunlight for the maximumoutput. Therefore, it causes inadequate performance in gaining energy. [What problem does your program solve? The description here should be in the problem space, not the solution space. —TPLT]

4.1.1 Terminology and Definitions

[This section is expressed in words, not with equations. It provide the meaning of the different words and phrases used in the domain of the problem. The terminology is used to introduce concepts from the world outside of the mathematical model The terminology provides a real world connection to give themathematical model meaning. —TPLT]

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 iteasier to correctly understand the requirements:

- Declination of the Sun: The angle between the rays of the Sun and the plane of the Earth's equator.
- Tilt angle: The angle for adjusting the solar panel result in the panel get the direct sunlight. [I think you mean that for optimum efficiency the tilt angle should equal the zenith angle, but the tilt angle is not always equal to the zenith angle. That is, you can have a solar panel that has a different tilt angle from the zenith angle. —SS][yes, it can. The actual tilt angle depends on the individual cases —Author]
- Date duration: The period of days depends on the input days of the users.
- Sun panel: The adjustable panel, which has solar cells on it, able to converse solar energy to power.
- Solar intensity: The amount of incoming solar energy, or radiation, that reaches the Earth's surface.
- Latitude: The angle which ranges from 0 ° at the Equator to 90° or -90° at the poles.
- Zenith Angle: The angle between the sun and the vertical. [Please see my comment under tilt angle. —SS][Ok —Author]

4.1.2 Physical System Description

[The purpose of this section is to clearly and unambiguously state the physical system that is to be modelled. Effective problem solving requires a logical and organized approach. The statements on the physical system to be studied should cover enough information to solve the problem. The physical description involves element identification, where elements are defined as independent and separable items of the physical system. Some example elements include acceleration due to gravity, the mass of an object, and the size and shape of an object. Each element should be identified and labelled, with their interesting properties specified clearly. The physical description can also include interactions of the elements, such as the following: i) the interactions between the elements and their physical environment; ii) the interactions between elements; and, iii) the initial or boundary conditions. —TPLT]

The physical system of SC, as shown in Figure: *ThePhysicSystem*, includes the following elements:

PS1: Solar panel, the panel with solar cells that able to absorb solar energy from the sun.

PS2: Sun, proving solar energy to the solar panel

[A figure here makes sense for most SRS documents —TPLT]

4.1.3 Goal Statements

[The goal statements refine the "Problem Description" (Section 4.1). A goal is a functional objective the system under consideration should achieve. Goals provide criteria for sufficient completeness of a requirements specification and for requirements pertinence. Goals will be refined in Section "Instanced Models" (Section 4.2.6). Large and complex goals should be decomposed into smaller sub-goals. The goals are written abstractly, with a minimal amount of technical language. They should be understandable by non-domain experts. —TPLT]

Given the user located latitude (Φ_P), the area of the solar panel (P_A), the day started to estimate angle ($year_{\text{Start}}$, $month_{\text{Start}}$, day_{Start}),the day when the estimation end ($year_{\text{End}}$, $month_{\text{End}}$, day_{End}), the goal statements are: [you should have spaces before opening brackets. Please see the document review checklists. —SS] [Fixed. —Author]

GS1: Predict the optimum tilt angle in the period of days (θ_T).

GS2: Predict the optimum produced solar energy in the period of days (P_E).

4.2 Solution Characteristics Specification

The instance models that govern SC 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.

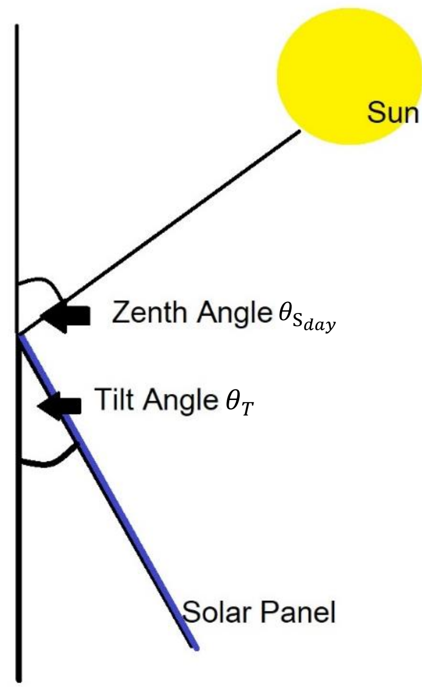

Figure 2: The Physic System

[This section specifies the information in the solution domai n 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 refiningmodels. 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]

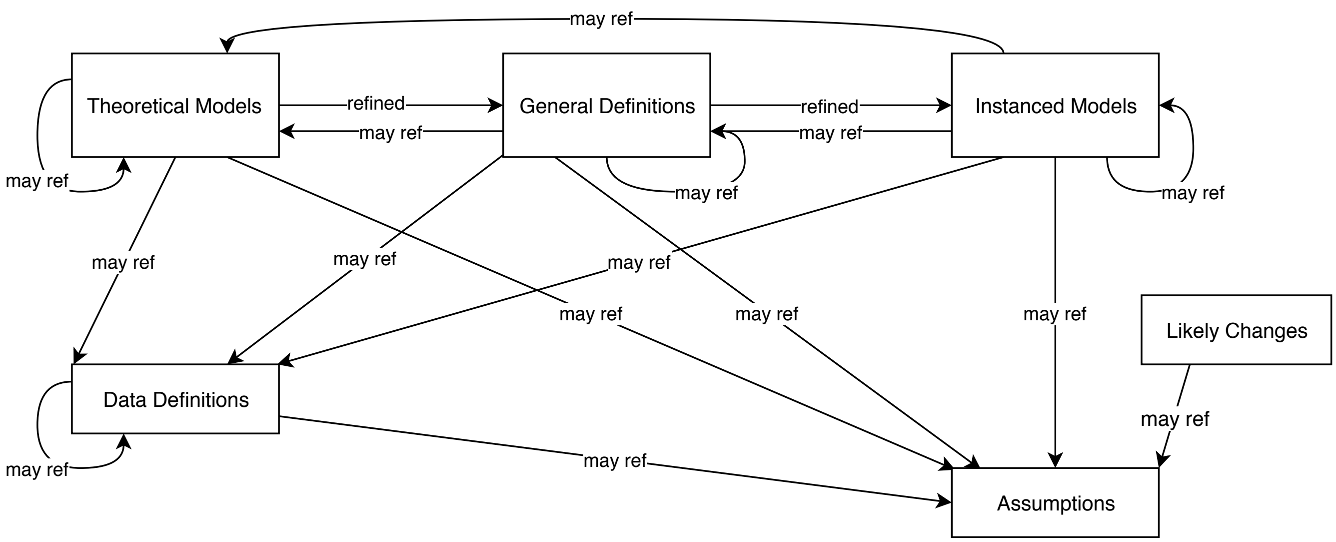

Figure 3: Models' Relations

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

4.2.1 Assumptions

[The assumptions are a refinement of the scope. The scope is general, where the assumptions are specific. All assumptions should be listed, even those that domain experts know so well that they are rarely (if ever) written down. —TPLT][The document should not take for granted that the reader knows which assumptions have been made. In the case of unusual assumptions, it is recommended that the documentation either include, or point to, an explanation and justification for the assumption. —TPLT]

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 environmentalcondition of the solar panel location assumed as users have an unobstructed viewof the sky, with no trees, hills, clouds, dust, or haze ever blocking the sun.(RefBy: GD1)

A2 : The default value of solar intensity(I_S) is provided in Section: Values of Auxiliary Constants.(RefBy: GD1)

A3 : This system calculate the solar zenith angle at noon.(RefBy: DD2)

A4 : Based on the resource of [6], this system definite the variable of P_r is refine by.(RefBy: IM3 and Values of Auxiliary Constants)

A5 : Based on the resource of https://photovoltaic-software.com/principle-ressources/how-calculate-solar-energy-power-pv-systems, this system definite the variable of PR is refine by.(RefBy: IM3 and Values of Auxiliary Constants)

A6 : Based on the resource of [7], the value of θ_T is refine by: IM1 and Output Variables

A7 : Based on the label of every solar panel, the value of P_E is refine by: IM3and Output Variables

[Short description of each assumption. Each assumption should have a meaningful label. Use cross-references to identify the appropriate traceability to T, GD, DD etc., using commands like dref, ddref etc. Each assumption should be atomic - that is, there should not be an explicit (or implicit) “and” in the text of an assumption. —TPLT]

[I suggest that you add a section for Type Definitions. You can see an example in the SRS by Deema (https://github.com/deemaalomair1/CAS741_project/tree/master/docs/SRS. Dealing with latitudes and longitudes is going to be confusing until you determine how they are represented. The following web-page has a good summaryof the representation of longitude and latitude: https://en.wikipedia.org/wiki/ISO_6709. My vote is to use positive and negative signs, instead of N, S, W, E. You can define North as positive and East as positive, then the sign gives you the direction. The numbers are for degree, minutes and seconds. Therefore, the coordinates could be represented using a type defined with a tuple. —SS] [You should also think about a type for dates. Again, a tuple makes sense. —SS]

4.2.2 Type Definition

This section defines the general data type that Sun Catcheruses.

day= a tuple of $(year, month, day)$

$date$ = a sequence of day

4.2.3 Theoretical Models

[Theoretical models are sets of abstract mathematical equations or axioms for solving the problem described in Section "Physical System Description" (Section 4.1.2). Examples of theoretical models are physical laws, constitutive equations, relevant conversion factors, etc. —TPLT]

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

Number	T1
Label	**Lambert's cosine law**
Equation	$I_0 = I \times \frac{\cos\theta \times \partial\Omega \times \partial A}{\cos\theta \times \partial\Omega_0 \times \partial A_0}$
Description	In the following source, Lambert's cosine law state that `''the radiant intensity or luminous intensity observed from an ideal diffusely reflecting surface or ideal diffuse radiator is directly proportional to the cosine of the angle $\theta$ between the direction of the incident light and the surface normal.''` [look at the writing checklist for a discussion of opening and closing quotation marks. —SS][ok —Author] I_0 is the radiance of illumination. I is the number of the photons. [When you show a symbol outside of an equation, it should be formatted as in the equation. In this case I. —SS][ok —Author] $\cos\theta \times \partial\Omega \times \partial A$ is the number of photonsper second emitted into the wedge at angle θ. where $\partial\Omega$ is an equal angle that represents each wedge in the circle. ∂A is the area element. $\partial\Omega_0$ is the portion of the observer's total angular field-of-viewof the scene. ∂A_0 is an aperture of area. [Again, look at the writing checklists. —SS][ok —Author]
Source	[Weik][8]
Ref. By	GD1

4.2.4 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 GDcould 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	**Calculating the Solar intensity**
SI Units	$\theta_{S_{\text{date}}}$
Equation	$I_{S_{\text{total}}} = I_S \cdot \frac{1.00}{I_S}^{sec(\theta_{S_{\text{date}}})}$
Description	This equation is calculating the total solar intensity in the date duration ($\theta_{S_{\text{date}}}$) is the zenith angle of the sun in the date in a squence of day. It is input from $DD2$. I_S is solar intensity. Its default value has described in the Assumption: A2.
Sources	https://www.solarpaneltilt.com/#other[7]
Ref. By	IM1

Derivation of simplified the calculation of total solar intensity

The above equation is driven by the $T1$, which is describing the calculation of the radiance of the illumination.
This formula assumes that the earth is flat, so a factor was applied to account for the curvature of the earth (and therefore the earth's atmosphere). These factors and the angle of the sun with respect to the panel represent as $\frac{1.00}{I_S}^{sec(\theta_{S_{\text{date}}})}$
Where $\frac{1}{\cos\theta_{S_{\text{date}}}}$ is $sec(\theta_{S_{\text{date}}})$

[I like that your T1 and GD1 are explicitly connected by your derivation. Good. —SS] [Thanks! —Author]

[The derivation shows how the IM is derived from the TMs/GDs. In cases where the derivation cannot be described under the Description field, it will be necessary to include this subsection. —TPLT]

4.2.5 Data Definitions

[The Data Definitions are definitions of symbols and equations that are given for the problem. They are not derived; they are simply used by other models. For instance, if a problem depends on density, there may be a data definition for the equation defining density. The DDs are given information that you can use in your other modules. —TPLT]

[All Data Definitions should be used (referenced) by at least one other model. —TPLT]

This section collects and defines all the data needed to build the instance models. The dimension of each quantity is also given. [Modify the examples below for your problem, and add additional definitions as appropriate. —TPLT]

Number	DD1
Label	**The Declination of the Sun**
Symbol	δ_{date}
SI Units	$^\circ$
Equation	-
Description	Declination of the Sun means the angle between the rays of the Sun and the plane of the Earth's equator. ``‘‘date’’`` is a sequence of day, which is a data type defined in Type Definition, and refers from DD3. The degree of δ_{date} can be determinated by Figure: Analemma.
Sources	[Holbrow][9] [Cengage.com][10],
Ref. By	DD2

[What is declination? You should repeat your definition from earlier in the document. —SS]

Detailed describetion of Analemma

``‘‘Analemma is often drawn on globes as a bigbottomed ‘‘figure 8,’’ shows the declination of the sun throughout the year. ’’``
The y-axis of analemma represent the degree of sun's declination, ``‘‘N’’`` denotes northern hemisphere and ``‘‘S’’`` denotes southern hemisphere.

In Sun Catcher, plus sign (+) denotes northern hemisphere or the equator and minus sign (-) denotes southern hemisphere.

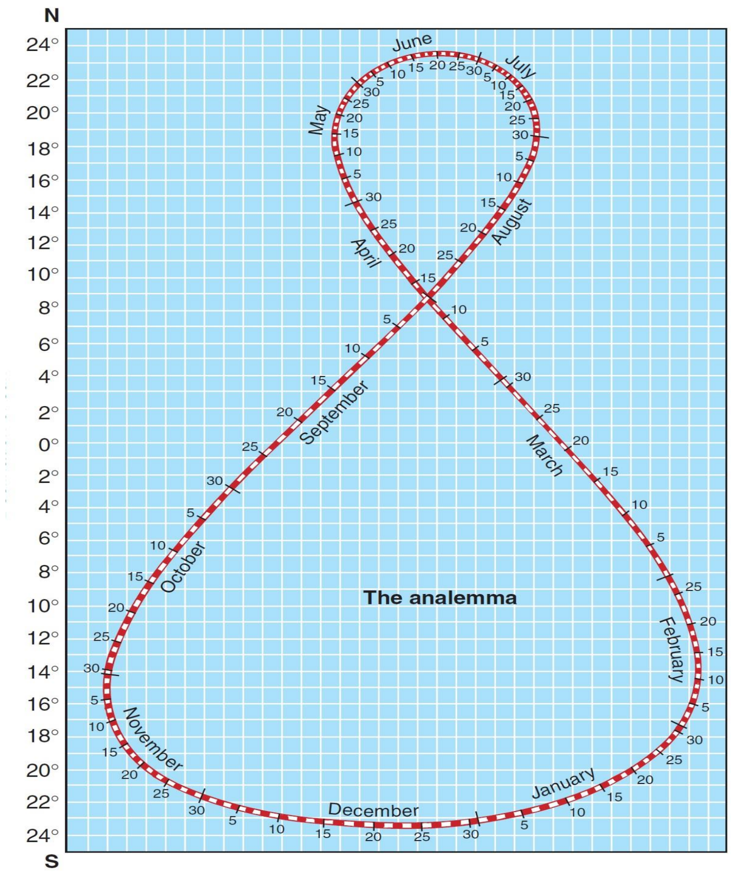

Figure 4: Analemma

Number	DD2
Label	**Calculate Zenith Angle**
Symbol	$\theta_{S_{\text{date}}}$
SI Units	$\circ$
Equation	$$\theta_{S_{\text{date}}} = \begin{cases} \Phi_P - \delta_{\text{date}}, & \text{if } 0 \leq \Phi_P, \delta_{\text{date}} \leq 90°\text{N} \ \vee \ 0 \leq \Phi_P, \delta_{\text{date}} \leq 90°\text{S} \\ \Phi_P + \delta_{\text{date}}, & \text{if } \begin{cases} 0 \leq \Phi_P \leq 90°\text{N} \ \wedge \ 0 \leq \delta_{\text{date}} \leq 90°\text{S} \\ 0 \leq \Phi_P \leq 90°\text{S} \ \wedge \ 0 \leq \delta_{\text{date}} \leq 90°\text{N} \end{cases} \end{cases}$$
Description	This equation describes the zenith angle associated with the solar panel. `''date''` is a sequence of day, which is a data type defined in Type Definition, and refers from DD3. $\theta_{S_{\text{date}}}$ is the zeith angle of the sun in the date of a squence of day. Φ_P is the local latitude of the solar panel from the users' input described at Goal Statements. δ_{date} is the declination of the vertical noon sun in the day. It is input from DD1
Source	[11]
Ref. By	GD1

Detailed derivation of simplified the calculation of zenith angle

According to the resource [11], it states the the solar elevation angle $\sin\alpha$ is determined by the relationship
$\sin\alpha = \sin\Phi \times \sin\delta + \cos\Phi \times \cos\delta \times \cos\text{h}$
where
α is the solar elevation.
Φ is the latitude.
δ is the solar declination.
h is the solar hour angle.
Based on the Level 1 celestial mechanics, the solar zenith angle is equal to 90° - the solar elevation angle
which can write as
$\cos\theta = \sin\alpha = \sin\Phi \times \sin\delta + \cos\Phi \times \cos\delta \times \cos\text{h}$ Based on the Assumptions section, it assumes that SC calculate the solar zenith angle at noon which mean solar hour angle $= 0°$.

Therefore, $\cos \mathrm{h} = 0$
Rewrite the equation as,
$\cos\theta = \sin\alpha = \sin\Phi \times \sin\delta + \cos\Phi \times \cos\delta \times 1$
$\Rightarrow \cos\theta = \sin\alpha = \sin\Phi \times \sin\delta + \cos\Phi \times \cos\delta$
Based on the Level 1 calcucltion, we have
$\cos\Phi \pm \delta = \sin\alpha = \sin\Phi \times \sin\delta + \cos\Phi \times \cos\delta$
Therefore, we have $\theta = \Phi \pm \delta$

Number	DD3
Label	**Calcuted the squence of the days**
Symbol	$date$
SI Units	-
Equation	$date = day_{Start} - day_{End}$
Description	day_{Start} is in the data type of day, which content a tuple $(year_{Start}, month_{Start}, day_{Start})$. day_{End} is in the data type of day, which content a tuple $(year_{End}, month_{End}, day_{End})$ $date$ is a squence that content the data type day The definition of data type can be found in Type Definition.
Sources	-
Ref. By	DD1

[Adding a data type for date would help make this clearer. —SS]

4.2.6 Instance Models

[The motivation for this section is to reduce the problem defined in "Physical System Description" (Section 4.1.2) to one expressed in mathematical terms. The IMs are built by refining the TMs and/or GDs. This section should remain abstract. The SRS should specify the requirements without considering the implementation. —TPLT]

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 4.2.5 to replace the abstract symbols in the models identified in Sections 4.2.3 and 4.2.4.

The goals Goal Statements are solved by IM1. [other details, with cross-references where appropriate. —TPLT] [Modify the examples below for your problem, and add additional models as appropriate. —TPLT]

Number	IM1
Label	**Calculating the Tilt Angle**
Input	$\theta_{S_{\text{date}}}, I_{S_{date}}$
Output	$\underset{\theta_{S_{\text{date}}}}{\arg\max} f(\theta_{S_{\text{date}}}) := \{I_S \cdot \frac{1.00}{I_{S_{\text{total}}}}^{sec(\theta_{S_{\text{date}}})} \vert \forall y : f(y) < f(\theta_{S_{\text{date}}})\}$ where$\theta_T = \theta_{S_{\text{date}}}, I_{S_{\text{max}}} = f(\theta_T)$ [write this using argmax `https://en.wikipedia.org/wiki/Arg_max`. —SS][I have tried this mathematical syntax but not should if it is correct expression —Author]
Description	Using General Definition: GD1 can get the total solar intensity $I_{S_{date}}$ during the period of date. Then using the value $I_{S_{date}}$ recalculates the optimum angle θ_T. $I_{S_{date}}$ is input from GD1. $\theta_{S_{\text{date}}}$ is input from DD1.
Sources	[Landau][7]
Ref. By	IM2

Number	IM2
Label	**Calculating the Daily Sun Intentisy**
Input	$\theta_{S_{\text{date}}}, I_{S_{\max}}$
Output	$I_{S_{\text{daily}}}$ $I_{S_{\text{daily}}} = I_S \cdot \frac{1.00}{I_{S_{\max}}}^{sec(\theta_{S_{\text{date}}})}$
Description	Using General Definition: GD1 can get the optimal solar intensity I_{S_T} of the date duration($date$). Then using the value I_{S_T} recalculates the optimum daily sun intensity $I_{S_{\text{daily}}}$. I_{S_T} is input from GD1. $\theta_{S_{\text{date}}}$ is input from DD1.
Sources	[Landau][7]
Ref. By	IM3

Number	IM3
Label	**Calculating the Solar Energy Output**
Input	P_A
Output	$P_E = P_A \times P_r \times I_{S_{\text{daily}}} \times PR$
Description	P is the solar panel. P_E is the estimated solar panel output energy. P_A is the solar panel area. It is input by the users P_r is solar panel yield or efficiency. Its default value has described in the Assumption: A4. $I_{S_{date}}$ is the solar intensity in a period of days of the noon. It is input from IM1 PR is the performance ratio, coefficient for losses. Its default value has described in the Assumption: A5.
Sources	[PhotovoltaicSoftware][12]
Ref. By	-

Derivation of simplify the solar energy output

The above equation is using the output from $GD1$, which is describingthe solar intensity. Some of the variable in this equation has its defult value, which describes in the section of *Assumptions*

4.2.7 Input Data Constraints

Table 1 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 theflexibility to experiment with unusual situations. The column of typical valuesis intended to provide a feel for a common scenario. The uncertainty column provides an estimate of the confidence with which the physical quantities can bemeasured. This information would be part of the input if one were performing anuncertainty quantification exercise.

The specification parameters in Table 1 are listed in Table 2.

Table 1: Input Variables

Var	Physical Constraints	Software Constraints	Typical Value	Uncertainty
Φ_P	$-90° \leq \Phi_P \leq 90°$	-	$(43°, 15'39.3")$	10%
$month$	$0 \leq month \leq 12$	-	12	10%
day	$0 \leq day \leq 31$	-	01	10%

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

Table 2: Specification Parameter Values

Var	Value
-	-

4.2.8 Properties of a Correct Solution

Table:*OutputVariables* 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.

Table 3: Output Variables

Var	Physical Constraints
θ_T	$0 \leq \theta_T \leq 90°$ (by A6)
P_E	$0 \leq P_E \leq$ Rated Maximum Power (by A7)

Rated Maximum Power is measured by manufacturers which shows the maximum power the solar panel can generate.

5 Requirements

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

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: Sun Catcher should able to read the input the values requested by the system from the system's interface which provides by users. The input values include the latitude of the solar panel, the date starts the calculation and the ending date.

R2: Sun Catcher should able to verify the input values by using the table of input's constrain: Input Variables.

[Requirements for the inputs that are supplied by the user. This information has to be explicit. —TPLT]

R3: Sun Catcher should able to output valid values for the problem defined by *ProblemDescription* in the corrected format.

R4: Using the input variables to generate the output values.

[It isn't always required, but often echoing the inputs as part of the output is a good idea. —TPLT]

R5: Sun Catcher should able to determine the tilt angle by using $IM1$

R6: Sun Catcher should able to determine the solar energy output by using $IM3$

[Calculation related requirements. —TPLT]

R7: Sun Catcher should able to verify the output by using the table of output's constrain: $OutputVariables$

[Verification related requirements. —TPLT]

R8: Sun Catcher should able to show the comparison between the outputs corresponding to the different user's input.

R9: The system output should remain consistent with the corresponding of the input values.

[Output related requirements. —TPLT]

[I would like to see a requirement that lets you compare different scenarios. The user should also be able to enter in a different set of inputs and see the results, along with the percent relative change in energy output compared to the first scenario. You'll want to include the specific formula for percent relative change. —SS] [Plan to show the difference of the outputs using a graphic and a table. —Author]

5.2 Nonfunctional Requirements

[List your nonfunctional requirements. You may consider using a fit criterion to make them verifiable. —TPLT] This section provides the non-functional requirements, the qualities that the software is expected to exhibit.

Correct: The outputs of the code have the properties described in Section: Properties of a Correct Solution

Verifiable: The code is tested with complete verification and validationplan

Understandable: The interface of this software shoulf support users to get the result within five stepts, and the output result should easily understand by the users.

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

6 Likely Changes

LC1: The system currently using the latest data of the solar's intensity, the calculation can be modified if this data upgrade in the future.$A3$

LC2: The system currently usingthe latest data of the solar panel's efficiency, the calculation can be modifiedif this data upgrade in the future.$A4$

LC3: The system currently using the latest data of the Performance Ratio, the calculation can be modified if this data upgrade in the future.$A5$

[Give the likely changes, with a reference to the related assumption (aref), as appropriate. —TPLT]

7 Unlikely Changes

LC4: The goal of the system is to predict the optimum tilt energy without considering the individual environmental difference from each case.A1

[Give the unlikely changes. The design can assume that the changes listed will not occur. —TPLT]

8 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 acomponent is changed, the items in the column of that component that are markedwith an "X" may have to be modified as well. Table 4 shows thedependencies of theoretical models, general definitions, data definitions, and instance models with each other. Table 5 shows the dependencies of instance models, requirements, and data constraints on each other. Table 6 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]

	T1	DD2	DD1	DD3	GD1	IM1	IM3
T1							
DD2			X				
DD1				X			
DD3							
GD1	X	X					
IM1					X		
IM3						X	

Table 4: Traceability Matrix Showing the Connections Between Items of Different Sections

	IM1	IM3	4.2.7	R2	R3	R6	R7	R??
IM1			X	X		X	X	X
IM3	X					X	X	X
R2	X		X		X			
R3	X	X						
R6	X	X						
R7	X	X						
R??	X	X			X			

Table 5: Traceability Matrix Showing the Connections Between Requirements and Instance Models

	A1	A2	A3	A4	A5	A6	A7
T1							
DD2			X				
DD1			X				
DD3							
GD1	X	X	X				
IM1		X				X	
IM3	X			X	X	X	X
LC1		X					
LC2				X			
LC3					X		
LC4	X						

Table 6: Traceability Matrix Showing the Connections Between Assumptions and Other Items

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

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 acomponent is changed, the components that it points to should also be changed. Figure 3 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.

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]

This section contains the standard values that are used for calculations in SC.

Symbol	**Description**	**Value**	**Unit**
I_S	Solar insensity	1.35	$\frac{k\text{W}}{m^2}$
P_r	Solar insensity	18.7	%
PR	Solar insensity	0.75	%

References

[1] W. Spencer Smith and Lei Lai. A new requirements template for scientific computing. In J. Ralyté, P. Ȧgerfalk, 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.

[2] W. Spencer Smith, Lei Lai, and Ridha Khedri. Requirements analysis for engineering computation: A systematic approach for improving software reliability. *Reliable Computing, Special Issue on Reliable Engineering Computation*, 13(1):83–107, February 2007.

[3] W. Spencer Smith. Systematic development of requirements documentation for general purpose scientific computing software. In *Proceedings of the 14th IEEE International*

Requirements Engineering Conference, RE 2006, pages 209–218, Minneapolis / St. Paul, Minnesota, 2006. URL http://www.ifi.unizh.ch/req/events/RE06/.

[4] W. Spencer Smith, John McCutchan, and Jacques Carette. Commonality analysis for a family of material models. Technical Report CAS-17-01-SS, McMaster University, Department of Computing and Software, 2017.

[5] Nirmitha Koothoor. A document drive approach to certifying scientific computing software. Technical Report CAS-17-01-SS, MA thesis. Hamilton, ON, Canada: McMaster University, 2013. URL https://macsphere.mcmaster.ca/bitstream/11375/13266/1/fulltext.pdf.

[6] Swiss Federal Laboratories for Materials Science and Technology (EMPA). Record efficiency of 18.7 percent for flexible solar cells on plastics, swiss researchers report. *ScienceDaily*, 2011. URL www.sciencedaily.com/releases/2011/05/110519101355.htm.

[7] Charles R. Landau. Optimum tilt of solar panels, 2001. URL https://www.solarpaneltilt.com/#other.

[8] Martin Weik. *Lambert's cosine law. In: Computer Science and Communications Dictionary*. Kluwer Academic Publishers 2000, Springer US., 1nd edition, 2000.

[9] Charles H. Holbrow. Build your own analemma. Technical Report 1302.0765v1, Colgate University and MIT, 2013. URL https://arxiv.org/pdf/1302.0765.pdf.

[10] Cengage.com. Sun angle, duration, and insolation. URL https://www.cengage.com/resource_uploads/downloads/0495555061_137179.pdf.

[11] Harold. M. Woolf. On the computation of solar elevation angles and the determination of sunrise and sunset times. Technical Report NASA-TM-X-1646, National Meteorological Center; Suitland, MD, United States, NASA Headquarters; Washington, DC, United States, 1968. URL https://ntrs.nasa.gov/search.jsp?R=19680025707.

[12] Photovoltaic software. How to calculate the annual solar energy output of a photovoltaic system? URL https://photovoltaic-software.com/principle-ressources/how-calculate-solar-energy-power-pv-systems.

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

- For Mac users `*.DS_Store` should be in `.gitignore`
- LaTeX 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 invokations 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]