

Your Wonderful Subsystem

Some One
Scientific Computing Division/Fermilab
Revision 1

Contents

1 Introduction	1
1.1 Purpose	1
1.2 Scope	2
1.3 Rationale	2
1.4 Terminology	2
2 Overview	2
3 Requirements	2
3.1 Actors	2
3.2 The Major Inputs and Outputs	2
3.3 Behavioral requirements (use cases)	2
3.4 Constraints	3
4 Architectural Overview	3
4.1 Roles	3
4.2 Functional unit or Component block diagram	3
4.3 Physical unit block diagram	3
4.4 Deployment scenario	4
5 Component Interfaces	4
6 Protocols	4
7 Discussion	4
7.1 Decisions and Choices	4
7.2 Architecture Rationale	4
7.3 Implications resulting from Choices	4
7.4 Resulting rules	4
7.5 Constraints imposed on other systems	5
8 Testing considerations	5

1 Introduction

1.1 Purpose

Describe what it is this report is going to convey or what this paper is about. Define the system that is being described and interesting qualities such as whether or not it is a utility to be shared by many. Include what this report should not include.

1.2 Scope

Indicate the boundaries of this subsystem and also for this report. This section includes information about what is covered in the report and examples of what is not.

1.3 Rationale

Why is this subsystem needed?

1.4 Terminology

Define the meaning of new words and acronyms along with how words are used in the context of this document.

2 Overview

Give a general overview of the basic function of this subsystem and how it fits into the world.

3 Requirements

Describe how this subsystem will be used by outside actors. Include constraints imposed by outside systems and any other important factors governing its design and implementation.

3.1 Actors

Describe here the set of characters that will be interacting with this system. This list should not include internal actors.

3.2 The Major Inputs and Outputs

Describe what the system needs to do its work and what it produces.

3.3 Behavioral requirements (use cases)

List the steps necessary to perform each important task associated with this system. Do not include any implementation details here, only brief statements of actions and responses for each one. Each use case should only cover one specific task. Below is a template for use cases.

3.3.1 Task name

Task: Name of this task.

Level: Summary/user goal/sub-function.

Goal: Stated as a short active verb phrase.

Actor: Who does this.

Trigger: Why this is happening.

Preconditions: Things which must exist before the use case can start, and any particular state the overall system must be in to allow performance of this case.

Post-conditions: New state which exists at successful completion of use case.

Description: Steps (numbered) to complete the task. This should include a narrative description of the manageable series of steps that make up the use case.

Nonstandard Flow: Exceptions (error conditions) to the standard flow or alternative success routes.

Comments: Link to other information regarding with use case.

3.3.2 Another task

Put in as many tasks as are needed.

3.4 Constraints

List any additional non-functional requirements here or reference them here. Examples include known external interfaces or protocols or performance constraints.

4 Architectural Overview

This section can contain a series of diagram illustrating parts of the subsystem and their relationship with other parts. Do all the following subsections make sense? Does it make sense to distinguish between a function or role and components?

4.1 Roles

The roles or functional units are defined and described in this section.

4.2 Functional unit or Component block diagram

Include here a picture of functional units within the system and their relationship to one another.

4.3 Physical unit block diagram

Show here the known hardware configuration that is to be built or is available for use.

4.4 Deployment scenario

How do functional units and components map to physical devices?

5 Component Interfaces

Expand the interfaces of the components shown in the previous section. Include relationships to other components here.

6 Protocols

Describe known elements of any protocol involved in data exchanges, external or internal to this subsystem, and the types of messages or data that may be exchanged.

Consider whether or not this should be distinguished from component interfaces.

Show important invocation or message exchange timing sequences here. Show what parts of the interfaces are used by other components.

7 Discussion

This section captures discussions and information that lead to the current architecture view and component organization.

7.1 Decisions and Choices

Other solutions that were considered and rejected should be briefly summarized here along with arguments for and against them. The purpose of doing this is so old arguments do not continually resurface.

7.2 Architecture Rationale

Why the current architecture and component interfaces are appropriate for the problem.

7.3 Implications resulting from Choices

Additional constraints that are imposed on the whole system or this system as a result of choices made here.

7.4 Resulting rules

Include all things that must be true in the system and rules that must be followed while the system is in operation. An example is that one worker node will only be assigned to one partition.

7.5 Constraints imposed on other systems

List what constraints this system imposed on other systems.

8 Testing considerations

Explain any load testing that must be performed to evaluate the performance of this system as a whole or parts within it. Suggestions for how to test (verification and validation) this system should be included. Ways of evaluating the performance of this system should be included here.

DRAFT