
SOFTWARE REQUIREMENTS SPECIFICATION

for

Scheduler simulation for the
Cranfield IT Departement

Version 1.0 approved

Prepared by Yoann Masson
December 15, 2017

Contents

1	Introduction	3
2	Overall Description	4
2.1	Product Perspective	4
2.2	Product Functions	4
2.3	User Classes and Characteristics	4
2.4	Operating Environment	4
2.5	Design and Implementation Constraints	5
2.6	User Documentation	5
2.7	Assumptions and Dependencies	5
3	External Interface Requirements	6
3.1	User Interfaces	6
4	System Features	7
4.1	User Management	7
4.1.1	Description	7
4.1.2	Settings from the input files	7
4.1.3	Functional Requirements	7
4.2	Ressource Management	8
4.2.1	Description	8
4.2.2	Settings from the input files	8
4.2.3	Functional Requirements	8
5	Other Nonfunctional Requirements	9
5.1	Performance Requirements	9
5.2	Safety Requirements	9
5.3	Security Requirements	9
5.4	Software Quality Attributes	9
5.5	Business Rules	10
6	Other Requirements	11
6.1	Appendix A: Glossary	11
6.2	Appendix B: Analysis Models	11
6.3	Appendix C: To Be Determined List	11

1 Introduction

Nowadays high powerfull computer is highly interesting for industries and research. To stay on top of the game, Cranfield University is purchasing a new super computer. It will be used by a large number of users (students, researchers, etc...), so a good ressources management and a good billing policy is mandatory. In order to achieve these goals, a simulation of the new job control system is requiered.

This document will stated the requirement specifications of the simulation of the new job control system. The software shall allow IT supports to run different simulations of the job scheduler for the new supercomputer and shall keep track of the accounting state at any point in time. The simulation shall model the behaviour of the computing platform to explore alternative accounting strategies. The requirements should be well explained and not subject to any interpretation.

2 Overall Description

2.1 Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

2.2 Product Functions

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

2.3 User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

2.4 Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

2.5 Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

2.6 User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

2.7 Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

3 External Interface Requirements

3.1 User Interfaces

The simulation will run on parameters set by users. All settings will be stored and can be changed in a provided file. When executing the simulation this file must be given as a parameter of the executable for the settings to be taken in account. Parameters will go from the number user to the accounting settings and the number/frequency of jobs. If no file is provided, default settings will be applied. Those default settings can be found as an appendix of this document.

The output of the simulation shall be printed on the console executing the simulation.

4 System Features

4.1 User Management

4.1.1 Description

Simulated users are the heart of the system since they are the one making job requests. They are dividing into three groups :

- students
- researchers
- IT Supports

Each user is entitled budget, IT Supports get unlimited budget while students have a budget depending on their Msc. Every researcher has financial ressources according to their groups, some of them can have additional ressources.

Student will be separate evenly between these 3 Msc: Computer, Water, Soil.

4.1.2 Settings from the input files

The settings that can be changed through the settings file are:

Table 4.1: User variable

Variable	Default Value	Description
STUDENT_NUMBER	60	the number of student considered in the simulation
RESEARCHER_NUMBER	20	the number of researcher considered in the simulation
COMPUTER_BUDGET	20.00	Budget attribut to one student from Computer Msc
WATER_BUDGET	10.00	Budget attribut to one student from Water Msc
SOIL_BUDGET	5.00	Budget attribut to one student from Soil Msc

4.1.3 Functional Requirements

Every variable must be greater or equal than 0

4.2 Ressource Management

4.2.1 Description

With this feature, the user can manage the power of the simulated computing system. The changes that can be made are about the numbers of nodes, the price of using these nodes and the cost for the IT departement for running the super computer.

4.2.2 Settings from the input files

The settings that can be changed through the settings file are:

Table 4.2: Computing variable

Variable	Default Value	Description
NODES_16_PROCESSOR_NUMBER	80	the number of traditional nodes with 16 processors considered
NODES_32_PROCESSOR_NUMBER	40	the number of traditional nodes with 32 processors considered
NODES_64_PROCESSOR_NUMBER	30	the number of traditional nodes with 64 processors considered
ACCELERATED_16_PROCESSOR_NUMBER	30	the number of accelerated nodes with 16 processors considered
ACCELERATED_32_PROCESSOR_NUMBER	20	the number of accelerated nodes with 32 processors considered
ACCELERATED_64_PROCESSOR_NUMBER	10	the number of accelerated nodes with 64 processors considered
SPECIALIZED_NUMBER	20	the number of specialized nodes considered
SMALL_PRICE	0.05	the price of one machine hour for computing small jobs.
MEDIUM_PRICE	0.05	the price of one machine hour for computing medium jobs.
LARGE_PRICE	0.05	the price of one machine hour for computing large jobs.
HUGE_PRICE	0.05	the price of one machine hour for computing huge jobs.
COMPUTER_COST	0.01	the cost for the IT Department per node&hour.

4.2.3 Functional Requirements

Every variable must be greater or equal than 0. There must be at least 128 nodes in total.

5 Other Nonfunctional Requirements

5.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

5.2 Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

5.3 Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

5.4 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

5.5 Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

6 Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

6.1 Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

6.2 Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

6.3 Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>