

Software Engineering Project (2IP40)

Project Group 1

Unit Test Plan

version 0.1.0 (Internally Accepted), 26 May 2006



Project Team:	Sven Bego	0550191
	Roel Coset	0548132
	Robert Leeuwestein	0546746
	Maarten Leijten	0547649
	Ivo van der Linden	0547632
	Joery Mens	0547515
	Marcel Moreaux	0499480
	Tim Muller	0547961
Project Manager:	Tom Kleijkers	0515015
Senior Manager:	L. Somers	TU/e HG 7.83
Advisor:	Y.Usenko	TU/e HG 5.71
Customer:	C. Plevier	Dutch Space
	H. de Wolf	Dutch Space

Abstract

This document describes the Unit Test Plan (UTP) for the SPINGRID project and was made according to the software engineering standard provided by the European Space Agency described in [ESA]. This document contains the description of the unit tests for the project. This project is one of seven assignments for the course 2IP40 at Eindhoven University of Technology.

Contents

1	Introduction	6
1.1	Purpose	6
1.2	Scope	6
1.3	List of definitions and abbreviations	7
1.3.1	Definitions	7
1.3.2	Abbreviations	8
1.4	Documents	8
1.4.1	Reference Documents	8
1.4.2	Applicable Documents	8
1.5	Overview	8
2	Test plan	9
2.1	Test items	9
2.2	Features to be tested	9
2.3	Test deliverables	9
2.4	Testing tasks	9
2.4.1	Before the test	10
2.4.2	During the test	10
2.4.3	After the test	10
2.5	Environmental needs	10
2.6	Test case pass/fail criteria	10
3	Test case specifications	11
4	Test procedures	12

5	Test report	13
----------	--------------------	-----------

Document Status Sheet

Document Title	Unit Test Plan
Document Identification	SPINGRID/Documents/product/UTP/0.1.0
Author(s)	R. Leeuwestein, R. Coset
Version	0.1.0
Document Status	draft / <u>internally accepted</u> / conditionally approved / approved

Version	Date	Author(s)	Summary
0.0.1	24-05-2006	R. Leeuwestein	Document creation
0.0.2	26-05-2006	R. Leeuwestein, R. Coset	First version for internal review
0.1.0	27-05-2006	R. Coset	Internally accepted

Document Change Report

Document Title	Unit Test Plan
Document Identification	SPINGRID/Documents/product/UTP/0.1.0
Date of Changes	N/A

10

Chapter 1

Introduction

1.1 Purpose

15 This document contains all information regarding the testing of the software units that together comprise the SPINGRID system. It describes how these units should be tested regarding their compliance to the [DDD], what the actual tests to be performed are and the results of running those tests. The tests should be written and executed during the coding phase of the project.

20 This document is intended for SPINGRID implementors, as they are the ones who should write and execute the unit tests and verify their results.

1.2 Scope

25 The software implements a computational grid. This grid is able to execute jobs when it receives an application accompanied by a set of data files. By hiding the complexity of grid technology the system will be easy to use. Usability is also increased by offering a web-based front-end for users to access the system.

1.3 List of definitions and abbreviations

1.3.1 Definitions

Agent	Application that is used by a resource provider to retrieve and execute jobs.
Application	A non-interactive data processing application consisting of executables, scripts and/or auxiliary data files that reads one or more input data files and writes one or more output files.
Application Provider	An application provider can offer a set of applications to the SPINGRID system. They can restrict access for projects and for resource providers to their applications.
Client	Application that is used by all the users except the resource provider who uses the agent application.
Computational Grid	A hardware and software infrastructure that enables coordinated resource sharing within dynamic organizations consisting of individuals, institutions and resources.
Customer	Dutch Space B.V.
Data Provider	A data provider can offer a set of datafiles to the SPINGRID system. They can restrict access for projects and for resource providers to their datafiles.
Dispatcher	A dispatcher acts like a server and manages the distribution of jobs over the computational grid.
Job	Specification of application, configuration data, input and/or output data files and scheduler specific data (priority, preferred resource, etc).
Job Provider	Job providers are users that offer a job to a project. They have to be a member of that particular project.
Project	A collection of jobs with specified access rights to which users (project members) can be assigned.
Project Administrator	The project administrators administrate projects and can assign and remove job providers, configure a project and restrict access for resource providers.
Resource Provider	Resource providers are users that offer time on their computers to the SPINGRID system. They can restrict access to their computer for application providers and projects.
Role	The actions and activities assigned to a person.
SPINGRID	A computational grid using SPINGRID software.
SPINGRID Software	Software developed by Dutch Space and TU/e to build computational grids for distributed data processing.
SPINGRID System	The full name of the entire system.
System Administrator	The system administrator oversees the entire SPINGRID system and has the right to configure the system, to create and remove projects and assign and remove project administrators.

30 1.3.2 Abbreviations

DDD	Detailed Design Document
ESA	European Space Agency
URD	User Requirements Document
UTP	Unit Test Plan

1.4 Documents

1.4.1 Reference Documents

[ESA]	<i>ESA Software Engineering Standards (ESA PSS-05-0 Issue 2)</i> , ESA Board for Software Standardization and Control (BSSC), 1991
[SVVP]	<i>Software Verification and Validation Plan</i> , SPINGRID team, TU/e, Version 0.1.2, March 2006

1.4.2 Applicable Documents

[ADD]	<i>Architectural Design Document</i> , SPINGRID team, TU/e, version 1.0.0., April 2006
[DDD]	<i>Detailed Design Document</i> , SPINGRID team, TU/e, version 0.0.1, April 2006
[URD]	<i>User Requirements Document</i> , SPINGRID team, TU/e, version 1.0.0, February 2006
[SRD]	<i>Software Requirements Document</i> , SPINGRID team, TU/e, version 1.0.1, March 2006

35 1.5 Overview

In the second chapter the items to be tested are mentioned. A specification for each test case is given in the third chapter. The fourth chapter specifies the procedures for these test cases. In the fifth chapter the reports for all test cases are presented.

Chapter 2

40 Test plan

2.1 Test items

The items to be tested are all of the SPINGRID classes. A list of the classes can be found in the [DDD]. For every class, all public methods should be tested.

2.2 Features to be tested

45 The classes as described in section 2.1 should be tested for compliancy to their postconditions, given their precondition. Both are enumerated in the [DDD].

2.3 Test deliverables

The following items must be delivered before the testing begins:

- The [DDD]
- The source code of each class
- Sections 1,2,3 and 4 of this document

The following items must be delivered when the testing is complete:

- The Unit Test Report (see section 5)
- Problem reports (if necessary)

55 2.4 Testing tasks

The testing procedure consists of three sets of tasks to be carried out: Those before the test, those during test and those after the test.

2.4.1 Before the test

Test cases must be written down, this includes input values and expected output values.

- 60 Because a class is not a stand alone program, but interacts with other classes, it might be necessary that a test program is written. The test program calls the methods of the module to be tested and contains stubs for all the other classes that are called by the class to be tested.

2.4.2 During the test

- 65 The output values are written down.

2.4.3 After the test

A test report is written. The report contains the test specification, the expected test values and the output values produced during the test.

2.5 Environmental needs

- 70 All tests should be run on the same type of system as that on which the different components of the SPINGRID software are designed to run on, see section 2.4 of the [SRD].

2.6 Test case pass/fail criteria

A test case passes when the produced output of the test program matches the expected output values and no errors have occurred. If this does not happen the test case fails.

⁷⁵ Chapter 3

Test case specifications

These can be found in the [DDD]

Chapter 4

Test procedures

- ⁸⁰ Since every class can be tested in the same way, no test procedures are defined. For every class, all tests for that class can be performed as described in section 2.4.

Chapter 5

Test report

Unit tests were successfully executed on 27-05-2006.