

Software Engineering Project (2IP40)

Project Group 1

Integration Test Plan

version 0.1.0 (internally accepted), 29 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 Integration Test Plan (ITP) 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 integration 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	11
2.4	Testing tasks	12
2.5	Environmental needs	12
2.6	Test case pass/fail criteria	12
3	Test case specifications	13
3.1	Integration test case I1	13
3.2	Integration test case I2	13
3.3	Integration test case I3	13
3.4	Integration test case I4	14
3.5	Integration test case I5	14

3.6	Integration test case I6	14
3.7	Integration test case I7	15
3.8	Integration test case I8	15
3.9	Integration test case I9	15
3.10	Integration test case I10	15
3.11	Integration test case I11	15
3.12	Integration test case I12	16
4	Test procedures	17
4.1	Integration test procedure TP1	17
5	Test report	18
5.1	TP1 Test Report	18
5.2	TP2 Test Report	18
5.3	TP3 Test Report	18

Document Status Sheet

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

	Version	Date	Author(s)	Summary
	0.0.1	22-05-2006	R. Leeuwestein	Document creation
	0.0.2	28-05-2006	J. Mens	Draft
5	0.1.0	29-05-2006	J. Mens	Internally accepted

Document Change Report

Document Title	Integration Test Plan
Document Identification	SPINGRID/Documents/product/ITP/0.1.0
Date of Changes	N/A

10

Chapter 1

Introduction

1.1 Purpose

This document describes the plans for testing the integration of the created components. The
15 purpose of this document is to test the interfaces between the components as described in
[ADD, chapter 5]. Every team member who cooperates in the integration tests should read
this document.

1.2 Scope

The software implements a computational grid. This grid is able to execute jobs when it
20 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.

1.3.2 Abbreviations

DDD	Detailed Design Document
ESA	European Space Agency
ITP	Integration Test Plan
URD	User Requirements Document

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

30 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

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

Test plan

2.1 Test items

The items to be tested consist of the integration of the code modules developed, for the SPINGRID project. For testing we choose the bottom-up approach. This means that integration testing starts at the bottom level. This way the project will be built up from the bottom level. The integration tests described in this documents are at the component level. The integration tests of lower level code modules are described in the corresponding components unit test. Unit tests are described in the [UTP].

2.2 Features to be tested

Figures 2.1, 2.2 and 2.3 shows the components that form the SPINGRID system. (these figures are derived from figures 4.1, 4.2 and 4.3 in the [ADD, chapter 4]). The arrows represent the order of integration, i.e. integration testing.

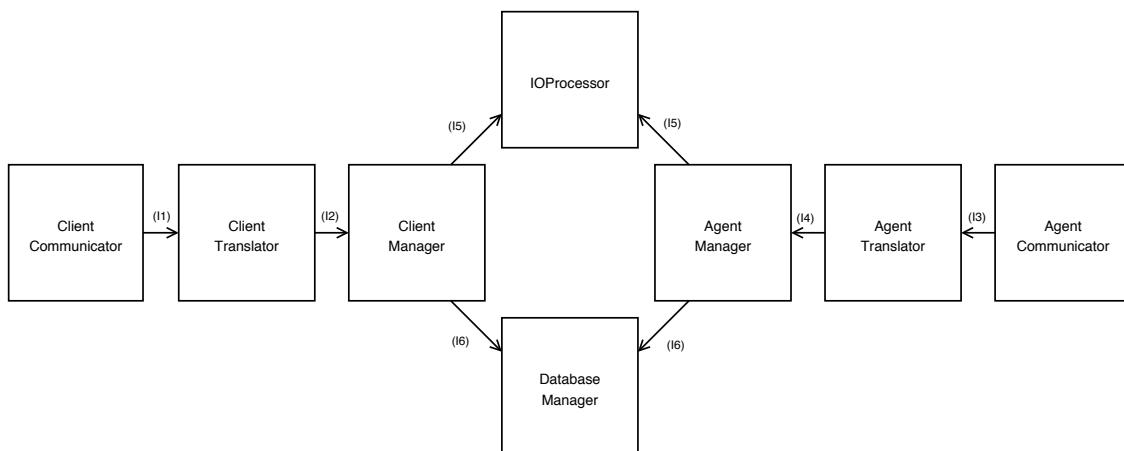


Figure 2.1: Components of the Dispatcher-software

50

Integration Tests of the dispatcher-software:

ID	Integration Test	Paragraphs		
I1	Client Communicator → Client Translator	3.1	4.1	5.1
I2	Client Translator → Client Manager	3.2	4.1	5.1
I3	Agent Communicator → Agent Translator	3.3	4.1	5.1
I4	Agent Translator → Agent Manager	3.4	4.1	5.1
I5	Agent Manager, Client Manager → Database Manager	3.5	4.1	5.1
I6	Agent Manager, Client Manager → IOProcessor	3.6	4.1	5.1

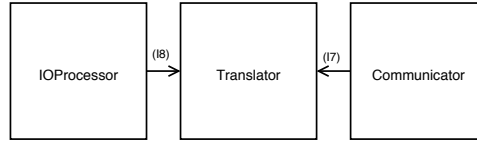


Figure 2.2: Components of the Client-software

55

Integration Tests of the client-software:

ID	Integration Test	Paragraphs		
I7	Communicator → Translator	3.7	4.2	5.2
I8	IOProcessor → Translator	3.8	4.2	5.2

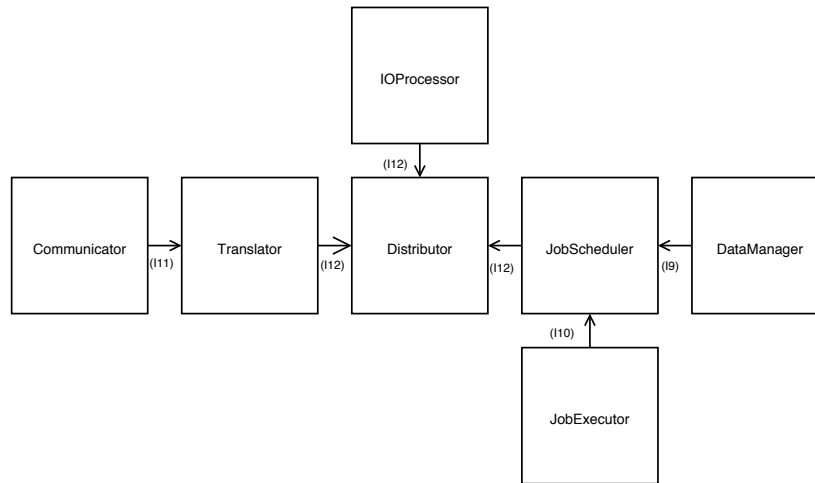


Figure 2.3: Components of the Agent-software

60

Integration Tests of the agent-software:

ID	Integration Test	Paragraphs		
I9	DataManager → JobScheduler	3.9	4.3	5.3
I10	JobExecutor → JobScheduler	3.10	4.3	5.3
I11	Communicator → Translator	3.11	4.3	5.3
I12	JobScheduler, IOProcessor, Translator → Distributor	3.12	4.3	5.3

65 2.3 Test deliverables

The following items must be delivered before integration testing can begin:

- The Architectural Design Document [ADD]
- Chapters 1 and 2 of the Detailed Design Document [DDD]
- The Unit Test Plan [UTP]
- 70 • Chapters 1, 2, 3 and 4 of this document

The following items must be delivered before a specific integration test can begin:

- The paragraphs in [DDD, chapter 3] concerning the components involved in the test
- The unit testing test reports concerning the components involved (i.e. these components have been unit tested). These reports can be found in [UTP, chapter 4]
- 75 • The components involved
- Drivers for this specific integration test (this is described below)
- Input data for this specific integration test.

A driver is a main program that accepts test data and passes this test to the component to be tested and prints relevant results. Drivers must be kept for future integration tests.

80 The storage of these files is described in the appendix for the DD phase of the [SCMP]. Because the two components involved in a specific integration test are already unit tested, the drivers that are made for them can be used in the integration test. This way testing can be performed more efficient. The following items must be delivered when a specific integration test is finished:

- 85 • Integration test report, which will be listed in the integration tests paragraph in chapter 5 of this document
- Integration test output data

- Problem reports (if necessary)

The following items must be delivered when testing on all code modules has finished:

- 90 • All integration test reports, which comprises chapter 5 of this document.

2.4 Testing tasks

The following tasks are necessary for performing a specific integration test:

- Designing the integration test
- Designing a driver (if it was not made at the unit test)

95 • Designing input test data (if it was not made at the unit test)

- Setting up a system, the components involved, the driver and the input test data
- Performing the integration test

2.5 Environmental needs

The environmental needs are described in [ATP, chapter 2.5].

100 **2.6 Test case pass/fail criteria**

Every test case must describe what the criteria are to pass that specific test.

Chapter 3

Test case specifications

3.1 Integration test case I1

Test Case Identifier	I1T1
Test Item(s)	Client Communicator → Client Translator
Input Specification	Create typical Client Communicator input
Output Specification	Check if the correct functions are called in the Client Translator
Environmental Needs	Client driver

3.2 Integration test case I2

Test Case Identifier	I2T1
Test Item(s)	Client Translator → Client Manager
Input Specification	Create typical Client Translator input A bit generic
Output Specification	Check if the correct methods are called in the Client Manager
Environmental Needs	I1 succeeded

3.3 Integration test case I3

Test Case Identifier	I3T1
Test Item(s)	Agent Communicator → Agent Translator
Input Specification	Create typical Agent Communicator input
Output Specification	Check if the correct methods are called in the Agent Translator
Environmental Needs	Agent Driver

3.4 Integration test case I4

Test Case Identifier	I4T1
Test Item(s)	Agent Translator → Agent Manager
Input Specification	Create typical Agent Translator input
Output Specification	Check if the correct methods are called in the Agent Manager
Environmental Needs	I3 succeeded

115

3.5 Integration test case I5

Test Case Identifier	I5T1
Test Item(s)	Client Manager → Database Manager
Input Specification	Create typical Client Manager input
Output Specification	Check if the correct methods are called in the Database Manager
Environmental Needs	I2 succeeded

Test Case Identifier	I5T2
Test Item(s)	Agent Manager → Database Manager
Input Specification	Create typical Agent Manager input
Output Specification	Check if the correct methods are called in the Database Manager
Environmental Needs	I4 succeeded

120

3.6 Integration test case I6

Test Case Identifier	I6T1
Test Item(s)	Client Manager → IOProcessor
Input Specification	Create typical Client Manager input
Output Specification	Check if the correct methods are called in the IOProcessor
Environmental Needs	I2 succeeded

125

Test Case Identifier	I6T2
Test Item(s)	Agent Manager → IOProcessor
Input Specification	Create typical Client Manager input
Output Specification	Check if the correct methods are called in the IOProcessor
Environmental Needs	I4 succeeded

3.7 Integration test case I7

Test Case Identifier	I7T1
Test Item(s)	Communicator → Translator
Input Specification	Create typical Communicator input
Output Specification	Check if the correct methods are called in the Translator
Environmental Needs	Dispatcher driver

130

3.8 Integration test case I8

Test Case Identifier	I8T1
Test Item(s)	IOProcessor → Translator
Input Specification	Create typical IOProcessor input
Output Specification	Check if the correct methods are called in the Translator
Environmental Needs	N/A

3.9 Integration test case I9

Test Case Identifier	I9T1
Test Item(s)	DataManager → JobScheduler
Input Specification	Create typical Datamanager input
Output Specification	Check if the correct methods are called in the JobScheduler
Environmental Needs	N/A

135

3.10 Integration test case I10

Test Case Identifier	I10T1
Test Item(s)	JobExecutor → JobScheduler
Input Specification	Create typical JobExecutor input
Output Specification	Check if the correct methods are called in the JobScheduler
Environmental Needs	N/A

3.11 Integration test case I11

Test Case Identifier	I11T1
Test Item(s)	Communicator → Translator
Input Specification	Create typical Communicator input
Output Specification	Check if the correct methods are called in the Translator
Environmental Needs	Dispatcher driver

140

3.12 Integration test case I12

Test Case Identifier	I12T1
Test Item(s)	JobScheduler → Distributor
Input Specification	Create typical JobScheduler input
Output Specification	Check if the correct methods are called in the Distributor
Environmental Needs	I9-I10 succeeded

145

Test Case Identifier	I12T2
Test Item(s)	IOProcessor → Distributor
Input Specification	Create typical IOProcessor input
Output Specification	Check if the correct methods are called in the Distributor
Environmental Needs	I8 succeeded

Test Case Identifier	I12T3
Test Item(s)	Translator → Distributor
Input Specification	Create typical Translator input
Output Specification	Check if the correct methods are called in the Distributor
Environmental Needs	I11 succeeded

150

Chapter 4

Test procedures

4.1 Integration test procedure TP1

Test Procedure Identifier	TP1
Purpose	This test procedure verifies whether the dispatcher software: <ul style="list-style-type: none">• can handle command-line input• can handle client input• can handle agent input• can output requested information to a client• can output requested information to an agent
Procedure Steps	Execute I5-I6 after I1-I4

Test Procedure Identifier	TP2
Purpose	This test procedure verifies whether the client software: <ul style="list-style-type: none">• can handle command-line input• can handle dispatcher input• can output requested information to the client• can output information to the dispatcher
Procedure Steps	Execute I7-I8

Test Procedure Identifier	TP3
Purpose	This test procedure verifies whether the agent software: <ul style="list-style-type: none">• can handle command-line input• can handle dispatcher input• can output information to the dispatcher
Procedure Steps	Execute I12 after I9-I11

Chapter 5

Test report

165 5.1 TP1 Test Report

I1 has successfully been executed by Robert Leeuwestein on May 29th, 2006.
I2 has successfully been executed by Robert Leeuwestein on May 29th, 2006.
I3 has successfully been executed by Joery Mens on May 29th, 2006.
I4 has successfully been executed by Joery Mens on May 29th, 2006.
170 I5 has successfully been executed by Joery Mens on May 29th, 2006.
I6 has successfully been executed by Joery Mens on May 29th, 2006.

5.2 TP2 Test Report

I7 has successfully been executed by Sven Bego on May 29th, 2006.
175 I8 has successfully been executed by Sven Bego on May 29th, 2006.

5.3 TP3 Test Report

I9 has successfully been executed by Roel Coset on May 29th, 2006.
I10 has successfully been executed by Roel Coset on May 29th, 2006.
180 I11 has successfully been executed by Roel Coset on May 29th, 2006.
I12 has successfully been executed by Roel Coset on May 29th, 2006.