Software Engineering Project (2IP40)

Project Group 1

Software Configuration Management Plan

version 0.1.2 (Internally Accepted), 14 June 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:	M. ter Linden	Dutch Space
	H. de Wolf	Dutch Space

Technische Informatica, Eindhoven University of Technology, Eindhoven

Abstract

This document is the Software Configuration Management Plan (SCMP) of the SPINGRID project. This project is part of the Software Engineering Project (2IP40) and is one of the assignments at Eindhoven University of Technology. The document complies with the SCMP from the Software Engineering Standard, as set by the European Space Agency [ESA].

This document contains information on the standards to be used for writing the documentation required for this project, as well as information about the processing and storage of these documents.

Contents

Document Status Sheet

Document Title	Software Configuration Management Plan
Document Identification	SPINGRID/Documents/Management/SCMP/0.1.2
Author(s)	J.C.J. Mens, M.L. Moreaux, M.C.G. Leijten, S.C.H. Bego
Version	0.1.2
Document Status	draft / internally accepted / conditionally approved / approved

Version	**Date**	**Author(s)**	**Summary**
0.0.1	16-12-2005	J.C.J. Mens, M.L. Moreaux, M.C.G. Leijten	Document creation
0.1.0	22-12-2005	J.C.J. Mens, M.L. Moreaux, M.C.G. Leijten	Internally approved
0.1.1	20-02-2005	M.C.G. Leijten	Appendix C added

Document Change Report

Document Title	Software Configuration Manager Plan
Document Identification	SPINGRID/Documents/Management/SCMP/0.1.2
Date of Changes	14-06-2006

Section Number	**Reason for Change**
Appendix C	This appendix wasn't defined yet
Appendices	Updated tools used in each phase

Chapter 1

Introduction

1.1 Purpose

The purpose of this document is to give guidelines and rules on the storage, layout and identification conventions of all documents that will be created in the course of the SPINGRID Software Engineering Project (2IP40), during the second and third trimester of the academic year 2005/2006 at the TU/e.

1.2 Scope

This document is about:

- Listing the specific documents which need to be written during the course of the project.
- Giving naming conventions for these documents.
- Providing a structured way to create, store and update the documents.
- Providing a standard layout for these documents.

This document will not describe the detailed contents of the individual documents.

The Configuration Items (CIs) that will be written during the SPINGRID project are:

- Architectural Design Document (ADD)
- Detailed Design Document (DDD)
- Software Configuration Management Plan (SCMP)
- Software Project Management Plan (SPMP)
- Software Quality Assurance Plan (SQAP)
- Software Requirements Document (SRD)
- Software Transfer Document (STD)
- Software User Manual (SUM)
- Software Verification and Validation Plan (SVVP)
- User Requirements Document (URD)
- Code
- Minutes and agenda
- Test plans for Unit Test (UT), System Test (ST), Integration Test (IT) and Acceptance Test (AT)
- Miscellaneous documents

1.3 List of definitions

ADD	Architectural Design Document
AT	Acceptance Test
CI	Configuration Item, any document that is considered a single entity
BCF	Bureau Computer Faciliteiten
CM	Configuration Manager
DDD	Detailed Design Document
ESA	European Space Agency
IT	Integration Test
QAM	Quality Assurance Manager
SCMP	Software Configuration Management Plan
SEP	Software Engineering Project (2IP40)
SPMP	Software Project Management Plan
SUM	Software User Manual
SQAP	Software Quality Assurance Plan
SRD	Software Requirements Document
ST	System Test
STD	Software Transfer Document
SVN	Subversion
SVVP	Software Verification and Validation Plan
TU/e	Technische Universiteit Eindhoven
URD	User Requirements Document
UT	Unit Test

1.4 List of references

[ADD]	*Architectural Design Document*, SPINGRID team, TU/e, Version 1.0.0, April 2006
[ATP]	*Acceptance Test Plan*, SPINGRID team, TU/e, Version 0.1.0, June 2006
[DDD]	*Detailed Design Document*, SPINGRID team, TU/e, not yet available
[ESA]	*ESA Software Engineering Standards (ESA PSS-05-0 Issue 2)*, ESA Board for Software Standardization and Control (BSSC), 1991
[ITP]	*Integration Test Plan*, SPINGRID team, TU/e, 0.1.0, May 2006
[SCMP]	*Software Configuration Management Plan*, SPINGRID team, TU/e, 0.1.2, June 2006
[SE1]	Software Engineering Website: Requirements for user requirements http://wwwis.win.tue.nl/2R690/ur_req.html, T. Verhoeff, TU/e, 2002
[SE2]	Software Engineering Website: Reviews http://wwwpa.win.tue.nl/wstomv/edu/sep/checklists/reviews.html, T. Verhoeff, TU/e, 2002
[SPMP]	*Software Project Management Plan*, SPINGRID team, TU/e, Version 0.1.1, January 2006
[SVVP]	*Software Verification and Validation Plan*, SPINGRID team, TU/e, Version 0.1.3, June 2006
[SQAP]	*Software Quality Assurance Plan*, SPINGRID team, TU/e, 0.1.3, June 2006
[SRD]	*Software Requirements Document*, SPINGRID team, TU/e, 1.0.1, March 2006
[STP]	*System Test Plan*, SPINGRID team, TU/e, 0.1.0, June 2006
[URD]	*User Requirements Document*, SPINGRID team, TU/e, 1.0.0, February 2006
[UTP]	*Unit Test Plan*, SPINGRID team, TU/e, 0.1.0, May 2006

Chapter 2

Management

2.1 Organisation

The roles directly involved in the configuration management are:

- Configuration Manager (CM)
- Assistant CM

2.2 Responsibilities

The Configuration Manager (CM) is mostly responsible for the technical part, such as the maintainance of the server.

Both the CM and his assistant are responsible for keeping the documentation up-to-date and correct. They may rename and move files that are incorrectly named or placed. However, they may not make significant changes to any project or product documents, and are mainly responsible for the naming and storing of these documents, not the actual contents. They are also responsible for making regular backups.

2.3 Interface management

There is a server located in E-hoog 11.13, TU/e and can be reached at the address `mir.luon.net`.

Bureau Computer Faciliteiten (BCF) has provided us with two computer systems to use in this project. When hardware problems occur with these machines BCF should be contacted. One of these machines will be used for backup archiving. See section 4.2.1.

2.4 SCMP implementation

Contrary to the ESA Software Engineering Standard [ESA] there won't be a separate SCMP document for each phase of the project. Instead this document will be updated with appendices for every phase of the project. For more information concerning planning of the phases, refer to the Software Project Management Plan (SPMP).

2.5 Applicable procedures

All the documents are subject to the standards described in the [ESA] and must also adhere to the requirements as described in the [SQAP] and the [SVVP].

To make a proper document the writer should apply to the following subsections.

2.5.1 Preamble and titlepage

Every document should start with defining which packages are used in the remainder of the document. Most of the time the following code will be enough:

```
\documentclass[12pt]{report}
\usepackage{a4wide}
\usepackage[english]{babel}
\usepackage{graphicx}
\usepackage{lineno}
```

The first line of code defines the style of the document and the default font size. The other lines define the packages as mentioned above. We use `a4wide` to use smaller margins, `graphicx` to be able to include images in the document and `lineno` to automatically generate linenumbers. Furthermore, we use English as the default language in our documents, so we include that package as well.

Then the following lines are to be included:

```
\input{../../global/Commandos.tex.inc}
\setlength\parskip{\bigskipamount}
\setlength\parindent{0pt}
\newcommand{\documenttitle}{DOCUMENT_NAME}
\newcommand{\documentdate}{DOCUMENT_DATE}
\newcommand{\documentversion}{DOCUMENT_VERSION}
```

in which `DOCUMENT_NAME` should be replaced by the actual name of the document, `DOCUMENT_DATE` should be replaced by the date on which the document is released and `DOCUMENT_VERSION` is the version number of the document described in section 3.1.4. These lines include a file named `Commandos.tex.inc` which includes some global commands like `\pn` for the project name. Furthermore, the marges are set and the three fields are set for the titlepage later on.

To include the titlepage you need to put the following line in the document right after `\begin{document}`:

```
\input{../../global/Titlepage.tex.inc}
```

2.5.2 Abstract

Each document should begin with an abstract summarizing the contents of the document. This can be achieved by `\begin{abstract} ... \end{abstract}`.

2.5.3 Contents

After the abstract, a table of contents in the form of a table summarizing all sections in the document should follow. We use `\tableofcontents` to generate this table.

2.5.4 Document status pages

The next sections should be the 'Document Status Sheet' and the 'Document Change Report'. The former will describe the current status of the document including its authors, title, identification, version and versionstatus (draft / internally accepted / conditionally approved / approved). The latter describes the changes made to the external approved document over time accompanied by the reasons of change. How this should be done can be seen at the beginning of this document which also includes a 'Document Status Sheet' and a 'Document Change Report'.

2.5.5 The rest of the document

The rest of the document should be structured using various sections and subsections as needed. If a lot of definitions are used it is wise to include a list of definitions somewhere in the first sections. The same goes for a list of references. A requirement which applies to all documents is that it should be written in English, except the minutes.

Chapter 3

Configuration identification

3.1 Naming conventions

3.1.1 General

All source files of the documents are kept in Subversion (SVN) (see section 4.1.1). The main LaTeX file for a document should be called:

- [document name].tex.

The naming scheme which should be followed for document names is described in section 3.1.3. When other files are included by the main LaTeX document, they should be called:

- [name].tex.inc for TEX files
- [name].[file type extension] for images.

Where name should be a short and clear name for the file. Documents generated from the LaTeX source which are intended for review are named:

- [document name]-[version tag].[file type extension]

3.1.2 Document IDs

All product and management documents carry a document ID, which consists of three parts concatenated by slashes ('/'):

1. The project name identifier, in this case SPINGRID.
2. The (relative) file path to the document's directory in the archive library (see also refArchiveLibrary) without the extension.
3. The version of the document.

3.1.3 Document names

Document names are required to abide by the following naming scheme:

- Product and management documents should be named after their abbreviation.
- Agenda should be named Agenda[DDMMYYYY], where YYYY should be replaced by the year, MM by the month and DD by the day the meeting is planned. For example the agenda for 12 January 2006 is called Agenda12012006.
- Minutes should be named Minutes[DDMMYYYY], which is analogous to the agenda.
- Progress reports should be named Progress[DDMMYYYY], which is analogous to the agenda.
- Project metrics should be named Metrics[DDMMYYYY], which is analogous to the agenda.

3.1.4 Version tags

The version tags follow the format x.y.z, where

- x denotes the external version number
- y denotes the internal version number
- z denotes the informal version number

A document's first version tag should be 0.0.1. The informal version number z of a document will be incremented whenever a new version of a document is created in order to let people give some informal feedback. The internal version y of a document will be incremented when a version of the document is internally approved. When this happens the informal version number z will be reset to zero. The external version x of a document will be incremented when a version of the document is externally approved. When this happens both internal version y and informal version z are reset to zero.

3.2 Baselines

Baselines are documents that have been internally reviewed and approved. They will be stored in the master library (see section 4.1.2)

According to the [ESA] new versions of the management documents need to be created for every stage of the project. Because of the small scale of this project it has been decided that the same management documents are used during the course of the project. Information specific for a stage in the project will be added to the documents in the form of appendices.

Chapter 4

Configuration control

4.1 Library control

All CIs are stored in one of three libraries: the development library, the master library or the archive library.

4.1.1 Development library

The development library contains CIs that are under construction and CIs that are not official product documents. Every category has its own SVN repository, in which the different documents have their own directory. The SVN directory tree as it exists on the server is shown in table 4.1. Two directories that may require a little explanation are tags and branches. The tags directory is used when tagging (a part of) the repository and the branches directory is used if a branch of the repository has to be created. The CM creates the initial module and group for each CI. Because of the nature of SVN, none of the users can overwrite or delete any of the files. The CM is allowed to correct naming and placing of documents if their location or name doesnt conform with the conventions described herein. When a document is ready for a review the authors can tag the SVN snapshot themselves. When a document has passed an internal or external review the authors must request the CM to tag the current SVN snapshot and to generate the document. In effect this means that authors are allowed to change the informal revision number by themselves, but that the CM is responsible for tagging the external and internal revisions.

```
/svn
  /sep
    /trunk
      /documents
        /global
        /management
          /SCMP
          /SPMP
          /SQAP
          /SVVP
        /meetings
          /client-meetings
          /group-meetings
        /product
          /DDD
          /SRD
          /URD
      /miscellaneous
      /presentations
      /source

  /tags
    /... Tagged documents
  /branches
```

Table 4.1: SVN directory tree

4.1.2 Master library

The master library contains CIs that have been internally approved. Only the CM can put CIs in the master library. CIs in the master library will never be deleted during the project. A copy of every document in the master library will also be kept on paper in the groups project room.

4.1.3 Archive library

The archive library contains CIs that have been externally released and approved. Only the CM can put CIs in the archive library. CIs may only be added after they have been externally reviewed and approved, as described in [SVVP] and the [SQAP] documents.

4.2 Media control

All documents are stored centrally on the server. This server is reachable under the domain name `mir.luon.net`

4.2.1 Backups

Every morning at 5:02 CE(S)T an backup is made on pcwin509.softeng.tue.nl and svgewis.win.tue.nl. Backup holds the following files:

- `/etc/` (Mir configurationfiles)
- `var/lib/mailman/archives/` (mailinglist archives)
- `var/lib/mailman/lists/` (mailinglist data)
- `/srv/www/` (website)
- `/srv/svn/` (svn repository with version control information)
- a fresh checkout of the entire svn repository

4.3 Change control

4.3.1 Development library

In the development library the change control system which SVN provides is used.

4.3.2 Master library

Once a CI is internally approved, the CM can put it in the master library. If authors want to make changes to a document inside the library, then that author has to contact the QAM. The QAM will call for a review meeting in which the changes are approved or rejected. More information regarding the change procedure can be found in [SVVP]. When the changes are accepted a new version of the CI will be put in the master library. The addition of appendices to a document does not require an additional review meeting, but can be done directly. The revision number should still be changed by the CM, however.

4.3.3 Archive library

CIs in this library cannot be modified under any condition. New versions may only be added after they have been externally reviewed and approved as described in the [SVVP]. As the CM is the only one allowed to create new documents in the archive library, there is no need for change control.

Chapter 5

Status accounting

All documents have a document change record in which all changes with respect to the previous version are recorded.

In the development library, SVN provides the status accounting.

Chapter 6

Tools, techniques and methods

All produced documents and tools must be freely available to any group member. Therefore there is a central storage facility that holds all files and can be accessed by all members.

This storage facility is the computer in E-hoog 11.13. From here on server will refer to this machine.

The server runs a number of services which the project members can access. For this access each member has been given a user account.

6.1 WWW

The server runs a webserver. The website for the group can be accessed by the project members on `https://mir.luon.net/`. The website contains the following information:

- Contact information of all the project members.
- A web-based interface for SVN.
- Read-only access to the master and the archive library.
- Links to relevant topics like the course page and SVN guides.

6.2 SVN

The SVN repositories are located on the server in the /srv/svn directory. The group members can access SVN at http://mir.luon.net/websvn using the secure http access method provided by most SVN clients. The advised tools for SVN are the subversion program for Linux users or Tortoise SVN for those running a Windows operating system. For documentation on using SVN see http://mir.luon.net/svn-book/svn-book.html.

6.3 LaTeX

All documents are written in LaTeX. For Windows MiKTeX can be used. A CD-ROM with MiKTeX is available in HG5.14.

6.4 Mailing list

For internal communications a mailinglist was created. The lists address is sep@mir.luon.net

Chapter 7

Supplier control

Refer to [SQAP] for the demands that are placed on tools supplied by external sources.

Chapter 8

Record collection and retention

Only the CM is allowed to delete documents from any library. All documents will be kept for the whole duration of the project.

Appendix A

SR phase

Additional tools that will be used during this phase:

- Dia 0.94, used to draw the UML diagrams.
- Yasper 1.0, used to draw petrinets.
- Eclipse 3.1, used to implement the prototype.

Appendix B

AD phase

Additional tools that will be used during this phase:

- Dia 0.94, used to draw several diagrams.

Appendix C

DD phase

Additional tools that will be used during this phase:

- Eclipse 3.1, used to implement the software.
- Fat Jar Eclipse plug-in, used to create an executable jar.
- Javadoc, used to extract comments from the code.