# Test Execution Document: Animator

DaCoPAn

Helsinki 11th April 2005

Software Engineering Project

UNIVERSITY OF HELSINKI
Department of Computer Science

UNIVERSITY OF PETROZAVODSK
Department of Computer Science

**Course**
581260 Software Engineering Project (6 cr)

**Project Group**
Carlos Arrastia Aparicio
Jari Aarniala
Alejandro Fernandez Rey
Vesa Vainio
Jarkko Laine
Jonathan Brown

Kirill Kulakov
Andrey Salo
Andrey Ananin
Mikhail Kryshen
Viktor Surikov

**Customer**
Markku Kojo

**Project Masters**
Juha Taina (Supervisor)
Yury Bogoyavlenskiy (Supervisor)

Turjo Tuohiniemi (Instructor)
Dmitry Korzun (Instructor)

**Homepage**
`http://www.cs.helsinki.fi/group/dacopan`

**Change Log**

| Version | Date | Modifications |
|---|---|---|
| 1.0 | 27.05.2004 | First version |

# Contents

# 1 Introduction

This document presents the results of the execution of the different tests presented in the DaCoPAn Animator test plan document, and other tests that the testing team has created in addition to those later. System testing results are presented in the common document "Test execution document: DaCoPAn", and results of the tests for the Analyzer module are presented in a separate document as well. The bug and fixes report can be found in the last section of the common document.

Section 2 introduces the results of unit testing. Section 3 presents the results of local integration testing. Section 4 explains the results of validation testing. Section 5 covers the results of other tests, which are explained.

The results of the test execution will be presented in subsection of this kind:

- Id and name of the executed test.
- Description of the test, or collected results.
- Errors found if any, and maybe a reference to the bug and fixes report.

Additionally to the presented format, the newly added tests include precisions on how the test was executed:

- Input for the test or steps to achieve it.
- Expected output of the test.

# 2 Unit testing

The test cases created during the testing phase and presented in the Test plan document were run on the Eclipse platform, which allows running JUnit test cases. The results were satisfactory as no errors or failures were highlighted by the test cases.

List of test completed:

- DataViewTestCase.java
- ENCTreeModelTestCase.java
- LayerViewTestCase.java
- NoteManagerTestCase.java
- ScenarioStepIteratorTestCase.java
- TransferUnitTestCase.java

- XMLProtocolEventsReaderTestCase.java
- ObjectSerializerTestCase.java
- ScenarioFileTestCase.java
- ScenarioSavingTestCase.java
- ENCPanelTest.java
- ExampleUITest.java
- MainFrameTestCase.java

# 3 Integration testing

This section presents the results of the local integration test cases achieved on the Animator module.

## 3.1 Test TC001 - ProtocolEventsReader and Dataview

- DESCRIPTION
  Test case TC001 (described in the Test Plan document) was executed succesfully, XMLProtocolEventsReaderTestCase JUnit test didn't highlight any errors.
- ERRORS FOUND
  No errors or failures were collected.

## 3.2 Test TC002 - Settings and user interface visualization

- DESCRIPTION
  Test case TC002 (described in the Test Plan document) was executed succesfully. All actions completed on the different setting options had the expected effect on the animation. The test was not only executed by running ExampleUITest.java but also by running the Animator module with real pefs and scenario files.
- ERRORS FOUND
  No errors or failures were collected.

## 3.3 Test TC003 - MSC buttons and encapsulation

- DESCRIPTION
  Test case TC003 (described in the Test Plan document) was executed succesfully.

All animation control buttons had the expected effect on the animation, and the encapsulation panel appeared as expected. The test was not only executed by running ExampleUITest.java but also by running the Animator module with real pefs and scenario files.

- ERRORS FOUND
  No errors or failures were collected.

## 3.4 Test TC004 - Localization

- DESCRIPTION
  Test case TC004 (described in the Test Plan document) was executed succesfully. The program changes all necessary features into the desired language.
- ERRORS FOUND
  Bugs 0020 and 0021 were collected, their explanation can be found in the common document "Test execution document: DaCoPAn", in its last section.

## 3.5 Test TC005 - Help menu

- DESCRIPTION
  Test case TC005 (described in the Test Plan document) was executed succesfully. Help menu is shown as expected.
- ERRORS FOUND
  No errors or failures were collected.

## 3.6 Test TC006 - Scenario and explore modes

- DESCRIPTION
  Test case TC006 (described in the Test Plan document) was executed succesfully. A more complex test achieved on the scenario mode is presented in last section "Other tests".
- ERRORS FOUND
  No errors or failures were collected.

## 3.7 Test TC007 - Time panel

- DESCRIPTION
  Test case TC007 (described in the Test Plan document) was executed succesfully. Time is shown correctly.
- ERRORS FOUND
  No errors or failures were collected.

## 3.8 Test TC008 - Note panel

- DESCRIPTION
  Test case TC008 (described in the Test Plan document) was executed succesfully. Notes are added, shown and deleted as expected.

- ERRORS FOUND
  No errors or failures were collected.

## 3.9 Test TC009 - Manually generated PEFs compliance to the XML DTD

- DESCRIPTION
  Test case TC009 (described in the Test Plan document) was executed succesfully. Manually generated pefs such as test_pef_01_enc.pef or test_pef_02_2layers.pef were loaded in the Animator succesfully and the visualized data was proved to be the same than the one contained in the pefs.

- ERRORS FOUND
  No errors or failures were collected.

## 3.10 Test TC010 - Analyzer generated PEFs compliance to the XML DTD

- DESCRIPTION
  Test case TC010 (described in the Test Plan document) was executed succesfully. PEFs generated by Analyzer were used for the most part of the testing phase. The pefs used and the results obtained can be found in the common document "Test execution document: DaCoPAn".

- ERRORS FOUND
  See bug report in last section of the above mentioned document.

## 3.11 Test TC011 - Loading a given tcpdump log pair into DaCoPAn software

- DESCRIPTION
  Test case TC011 (described in the Test Plan document) was executed succesfully. Results obtained can be found in the common document "Test execution document: DaCoPAn", in section 3 "Other logs".

- ERRORS FOUND
  See results presented in that section and bug report in last section of the above mentioned document.

### 3.12 Test TC012 - Loading a network scenario tcpdump log pair into DaCoPAn software

- DESCRIPTION
  Test case TC012 (described in the Test Plan document) was executed succesfully. Results obtained can be found in the common document "Test execution document: DaCoPAn", in section 2 "Networking scenario tcpdump log pairs".
- ERRORS FOUND
  See results presented in that section and bug report in last section of the above mentioned document.

## 4 Validation testing

This section presents the results of the validation test cases achieved on the Animator module.

### 4.1 Test TC013 - User requirements

- DESCRIPTION
  Test case TC013 (described in the Test Plan document) was executed succesfully. Extensive comparison between the Animator module features and behaviour and the user requirements presented in the DaCoPAn Requirements specification document proved to satisfy these.
- ERRORS FOUND
  No errors found.

### 4.2 Test TC014 - Use cases

- DESCRIPTION
  Test case TC014 (described in the Test Plan document) was executed succesfully. In a similar way than to the test case TC0013, use cases presented in the DaCoPAn Requirements specification document were proved to be satisfied.
- ERRORS FOUND
  No errors found.

### 4.3 Test TC015 - System requirements

- DESCRIPTION
  Test case TC015 (described in the Test Plan document) was executed succesfully.

Playing with the Animator different feature revealed a considerable amount of bugs that were corrected dynamically throughout the testing phase.

- ERRORS FOUND
  See bug report.

# 5 Other tests

This section presents other tests achieved on the Animator module and the results of the execution of those.

## 5.1 Moving now line

- INPUT/ STEPS
  Load an animation sequence file or a protocol events file. At any point, click on the MSC panel or drag the now line.
- EXPECTED OUTPUT/ BEHAVIOUR
  The now line should move to the point where the user clicked, showing the protocol data above the now line.
- DESCRIPTION OF THE EXECUTION
  The manual execution of the test proves that the feature works correctly.
- ERRORS FOUND
  No errors found.

## 5.2 Scenario recording

- INPUT/ STEPS
  - 1 - Run Animator and load a valid pef. Then Press Animation->Scenario mode in the upper menu.
  - 2 - Press "Recording mode" button ("Next item" button should not have any effect).
  - 3 - Press "Record start" and then play button in toolbar or just play button in toolbar.
  - 4 - "Record end" records the end time for a given animation item (MSC).
  - 5 - If the encapsulation button is enabled in UFO, press it.
  - 6 - Select any MSC item of the playlist, and press "Edit item" button (No effect on ENC items).

  - 7 - Press "Insert pause" button.
  - 8 - Notes can be added/edited/deleted at any point following the usual procedure.
  - 9 - Press save button in toolbar, or use File->Save or File->Save as in the upper menu.

- EXPECTED OUTPUT/ BEHAVIOUR

  - 1 - The scenario playlist panel should appear, with two available buttons "Recording mode" and "Next item".
  - 2 - This turns on the recording mode inside scenario mode. New buttons should appear: "Insert encapsulation" (disabled probably), "Edit item", "Record start", "Delete item", "Record item" (disabled probably), "Insert pause". The only buttons with effect should be "Record start" and "Insert pause". Pushing again "Recording mode" turns on play mode.
  - 3 - Pressing "Record start" button adds a new item to the playlist, similar to "MSC 0,0000->*[Network]1/1". Pressing play button, in addition to this, starts the animation.
  - 4 - The item should change and be similar to: "MSC 0,0000->X[Network]1/1", where X is the timestamp for the end time, which has been added by pressing the button.
  - 5 - Encapsulation diagram should appear on screen, and encapsulation item should be added to the playlist. "Insert encapsulation" button in the scenario playlist panel should allow you to add several items of the same encapsulation (with different comments, for instance).
  - 6 - A panel should appear, allowing the user to change the settings for that animation item (Scale settings, variables and protocol fields shown and autoplay feature).
  - 7 - Pause item should be added to the playlist.
  - 8 - See test case TC008 to check Note panel.
  - 9 - The animation sequence playlist should be saved into an animation sequence file (*.SCE).

- DESCRIPTION OF THE EXECUTION
  The manual execution of the test proves that the feature works correctly.

- ERRORS FOUND
  No errors found.

## 5.3 Scenario playing

- INPUT/ STEPS
  Load a valid animation sequence file. After pressing play, the playlist should be played. If the played item is encapsulation, the animation will wait until the user presses the "Quit encapsulation" button. If the next item after encapsulation is MSC but the autoplay option is disabled, the user will have to press play again.

- EXPECTED OUTPUT/ BEHAVIOUR
  The expected results are those explained in the previous point.

- DESCRIPTION OF THE EXECUTION
  The manual execution of the test proves that the feature works correctly.

- ERRORS FOUND
  No errors found.