Eindhoven University of Technology

OpenACCEL

Software Transfer Document

Authors:
Edward Brinkmann (0720803)
Carl van Dueren den Hollander (0743652)
Philippe Denissen (0719003)
Kevin de Ridder (0745581)
Leo van Gansewinkel (0781263)
Loc Tran (0766987)
Roel Jacobs (0767494)
Roy Stoof (0767157)
Jacco Snoeren (0772658)

Project Manager:
Erikos Alkalai

Quality Manager:
Duc Luu

Senior Management:
Lou Somers

Advisor:
Erik Scheffers

Customer:
Kees van Overveld

June 27, 2014

Abstract

This is the Software Transfer Document (STD) for the OpenACCEL Software Engineering Project. This document is based on the ESA standard for software development. This document complies with the ADD from the Software Engineering Standard, as specified by the European Space Agency (ESA) [11]. This document describes the procedures for the transfer of OpenACCEL to the customer.

Contents

Document Status Sheet

Document Status Overview

General

Document title:	Software Transfer Document
Identification:	STD1.0.pdf
Author:	T.L. Tran & J.P.H. Snoeren
Document status:	Draft

Document History

Version	*Date*	*Author*	*Reason of change*
0.1	June 23, 2014	T.L. Tran	Initial version
0.2	June 24, 2014	T.L. Tran	Changes made to the build procedure, added installation notes for Node and Git
0.3	June 25, 2014	T.L. Tran & J.P.H. Snoeren	Added chapter 5
0.4	June 27, 2014	T.L. Tran & J.P.H. Snoeren	Added chapter 6, 7 and 8
1.0	June 27, 2014	T.L. Tran & J.P.H. Snoeren	Internally approved

Document Change Records since previous issue

General

Datum:	June 27, 2014
Document title:	Software Transfer Document
Identification:	STD1.0.pdf

Changes

Page	*Paragraph*	*Reason to change*
5	2.x	Unclear instructions have been made m
7	3	Added installation notes for Node and
9	Chapter 5	Added UCRs that are not covered in t tance test.
10,11,12	Chapter 6,7,8 * Added reports of first acceptance test.	

Chapter 1

Introduction

1.1 Purpose

This document gives all the information concerning the transfer of the product from the OpenACCEL group to the customer. It describes the procedures to build OpenACCEL and procedures to access OpenACCEL by the users. It describes the items to be transferred and how the product has been tested prior to the transfer. Finally, it describes the state of these items compared to the requirements from the URD

1.2 Scope

OpenACCEL is a web-based application designed and developed by group OpenACCEL for Kees van Overveld at the Eindhoven University of Technology. The application is designed to aid students to get an understanding how to model with secondary educational mathematical knowledge.

1.3 List of Definitions & Abbreviations

1.3.1 Definitions

OpenACCEL	The new and improved application which will be developed by team OpenACCEL, based on ACCEL
NodeJS	A JavaScript platform for easily building fast, scalable network applications
Git	Version Control System

1.3.2 Abbreviations

URD	Architectural Design Document
SRD	Software Requirements Document
ADD	Architectural Design Document
DDD	Detailed Design Document
SUM	Software User Manual

STD	Software Transfer Document
SVVP	Software Validation and Verification Plan
UTP	Unit Test Plan
ITP	Integration Test Plan (ITP)
ATP	Acceptance Test Plan (ATP)
NPM	NodeJS Package Manager

1.4 References

[1] OpenACCEL, User Requirements Document (URD), version 1.4, May 2014.

[2] OpenACCEL, Software Requirements Document (SRD), version 1.0, June 2014.

[3] OpenACCEL, Architectural Design Document (ADD), version 1.0, June 2014

[4] OpenACCEL, Detailed Design Document (DDD), 1.0, June 2014

[5] OpenACCEL, Software User Manual (SUM), 1.0, June 2014

[6] OpenACCEL, Software Transfer Document (STD), 1.0, June 2014

[7] OpenACCEL, Software Validation and Verification Plan (SVVP), 1.0, June 2014

[8] OpenACCEL, Unit Test Plan (UTP), 1.0, June 2014

[9] OpenACCEL, Integration Test Plan (ITP), 1.0, June 2014

[10] OpenACCEL, Acceptance Test Plan (ATP), 1.0, June 2014

[11] ESA PSS-05-0 Issue 2, Software requirements and architecture engineering process, February 1991.

1.5 Overview

The remainder of this document gives the build and installation procedures in chapter 2. Chapter 3 gives the procedure to install OpenACCEL. All the customer items which are transferred the customer are given in chapter 4 and chapter 5 gives statistics of the acceptance tests. Chapter 6 describes problems found during the acceptance test OpenACCEL. Chapter 7 describes changes that are requested based on the acceptance test. Chapter 8 describes modifications made based on the requested changes in chapter 7.

Chapter 2

Build Procedure

OpenACCEL in general does not need building, since OpenACCEL is interpreted. However, when the project is downloaded from `GitHub`, it is necessary to move the files to the correct folder. The JavaScript documentation of OpenACCEL has to be build. In the following sections we will describe the procedure to set up the environment in which OpenACCEL can be build, how to retrieve the most up-to-date version of OpenACCEL from `GitHub` and how to build the respective parts.

2.1 Configuring environment

To retrieve and build OpenACCEL we need to install Node and Git. In this section we will describe the procedures to install these and how we can retrieve the code. Note: when running the scripts, `Linux` users should run the scripts ending with the extension `.sh`. Users using Windows should run scripts ending with the extension `.bat`. The scripts can be found in the main folder of the delivered files.

1. Copy everything from the CD to a local folder.
2. Install `Node`
   (a) `Linux`
      i. Install `Node` from your repository.
      ii. *Alternatively* install `Node` from `install/linux` folder on the CD. Refer to section 3.1.2 for installation notes.
      iii. Install `git` from your repository.
      iv. *Alternatively* install `git` from the `install/linux` folder on the CD. Refer to section 3.2.2 for installation notes.
   (b) `Windows`
      i. Install Node with the installer from the official website `http://nodejs.org/`.
      ii. *Alternatively* install `Node` from the `installers` folder on the CD. Run `node-v0.10.29-x86` if your computer is 32 bits and `node-v0.10.29-x64` if your computer is 64 bits. Refer to section 3.1.1 for installation notes.
      iii. Install `Git Bash` from `http://git-scm.com/`

iv. *Alternatively* install `Git Bash` from the `install/windows` folder on the CD. Refer to section 3.2.1 for installation notes.

3. The CD contains the sources of the latest version up until June 27, 2014, to get the latest version follow:

   (a) run the script: `Get Sources`

   (b) run the script: `Download Dependencies`

2.2 JavaScript documentation

To build the documentation run the script : `Build HTML documentation`. The documentation can be found in the `doc` folder in `HTML` format.
Alternatively to build the documentation into a LaTeX-document run the script : `Build LaTeX documentation` . The documentation can be found in the `doc-latex` folder in LaTeX-format.

2.3 Server

To build OpenACCEL run: `Build Sources`. The application can be found in the `bin` folder. One can run the script `Run Server` to run OpenACCEL directly after building. Accessing OpenACCEL can be done by visiting `localhost:8080` with an appropriate browser mentioned in the ADD [3].

2.4 Webserver

To be able to host OpenACCEL a web server is required, such as `Apache, Lighttpd` or `Nginx`. We assume that an appropriate web server has been installed on the server and thus, no instructions will be given on how to install a web server.

After building OpenACCEL the whole application can be found in the `bin` folder, these files will have to be moved to the appropriate folder depending on the web server.

2.5 Testing

One is able to test the project by running the script: `Run tests`.

2.6 Client

OpenACCEL does not require any additional building from the client. The client has to have a supported browser which can be found in the ADD [3]. The client can access OpenACCEL via the `url` of the web server.

Chapter 3

Installation Procedure

This section contains notes regarding procedures for installing `Node` and `Git`.

3.1 Node

3.1.1 Windows

The Windows installation of `Node` is basic, the basic configuration in the installation should not be changed. Thus, it suffices to click `Next` until the installation has finished.

3.1.2 Linux

Installing dependencies of `Node` requires an Internet connection, due to the large number of Linux distributions. Thus, an offline installation of `Node` is not trivial. As a reference the latest sources as of June 27, 2014have been added to the CD.

3.2 Git

3.2.1 Windows

The Windows installation of `Git Bash` is basic, the basic configuration in the installation should not be changed. Thus, it suffices to click `Next` until the installation has finished.

3.2.2 Linux

Installing dependencies of `Git` requires an Internet connection, due to the large number of Linux distributions. Thus, an offline installation of `Git` is not trivial. As a reference the latest sources as of June 27, 2014have been added to the CD.

Chapter 4

Configuration Itemlist

The following documents will be delivered:

- URD [1]
- SRD [2]
- DDD [4]
- SUM [5]
- STD [6]
- SVVP [7]
- UTP [8]
- ITP [9]
- ATP [10]

All code present in the `GitHub` repository will be delivered. This represents a fully working system.

Chapter 5

Acceptance Test Report Summary

See the ATP [10] for a detailed description of the Acceptance tests. These acceptance tests cover the user requirements in the URD [1], except for the user requirements that are not implemented. These are listed in Table 5.1. UCR 84-90, UCR94 and UCR95 are constraint requirements which cannot be tested.

UCR2	UCR44	UCR56	UCR70	UCR78
UCR3	UCR45	UCR57	UCR71	UCR81
UCR22	UCR46	UCR58	UCR72	UCR82
UCR37	UCR47	UCR59	UCR73	UCR83
UCR40	UCR48	UCR60	UCR74	UCR96
UCR41	UCR53	UCR67	UCR75	
UCR42	UCR54	UCR68	UCR76	
UCR43	UCR55	UCR69	UCR77	

Table 5.1: Non-implemented user requirements

Chapter 6

Software Problem Reports

The following issues occurred during the first acceptance test.

- Entering a syntax error in the I/O Edit tab does not display an error message, but deletes the script instead.
- Entering 'slider' (or any other input function) as a quantity on the right hand side without declaring it first gives the value NaN.
- When switching between tabs, error messages do not disappear.
- The history operator causes performance issues. The history should not be saved permanently, but only for as long as it is needed.
- If a script is time dependent, we have set the number of iterations to be executed and we optimize it using Genetic Optimization, the same number of generations should be executed for each generation.
- The genetic optimization tab should delete its data after recompilation.

Chapter 7

Software Change Requests

- When clicking on a quantity in the dependencies list, its definition is displayed in the input field and dependencies are changed accordingly.
- When clicking on a quantity in the todo list, the name of the quantity together with '=' is displayed in the input field.
- The results list should have a fixed width.
- The functions getChan() and putChan() should work.

Chapter 8

Software Modification Reports

We implemented fixes for all of the changes described in chapter 6 and chapter 7.