Test Plan
Diagnosis of ADAPT system

Version 1.0

Author: Emil Nilsson
Date: December 4, 2009

Status

Reviewed		09-10-14
Approved		09-10-14

Course name:	Control Project	E-mail:	diagnos2009@googlegroups.com
Project group:	FFF	Document responsible:	Emil Nilsson
Course code:	TSRT10	Author's E-mail:	emini550@student.liu.se
Project:	Diagnosis	Document name:	Testplan1_0.pdf

Project Identity

Group E-mail:	diagnos2009@googlegroups.com
Homepage:	http://www.isy.liu.se/edu/projekt/tsrt10/2009/
Orderer:	Erik Frisk, Linköping University
	Phone: +46 (0)13 28 2035 , **E-mail:** frisk@isy.liu.se
Customer:	The Division of Vehicular Systems, Linköping University
	Phone: +46 (0)13 28 1000 , **E-mail:** Vehicular.Systems@isy.liu.se
Course Responsible:	David Törnqvist, Linköping University
	Phone: +46 (0)13 28 1882, **E-mail:** tornqvist@isy.liu.se
Project Manager:	Niklas Wahlström
Advisors:	Mattias Krysander, Linköping University
	Phone: +46 (0)13 - 28 2198 , **E-mail:** matkr@isy.liu.se

Group Members

Name	Responsibility	Phone	E-mail
Niklas Wahlström	Project manager	0705-122349	nikwa148@student.liu.se
Daniel Eriksson	Document manager	073-4405730	daner963@student.liu.se
Erik Almqvist	Software manager	0705-149935	erija952@student.liu.se
Emil Nilsson	Test manager	073-6766558	emini550@student.liu.se
Andreas Lundberg	Design manager	0704-061227	andlu549@student.liu.se

Test Plan

Version	Date	Changes made	Sign	Reviewer
0.1	09-10-12	First draft.	OK	Emil Nilsson
1.0	09-10-14	First release.	OK	Emil Nilsson

Contents

1 Introduction

The purpose of this document is to specify how the requirements made in the Requirement List**?**, for the project Diagnosis of ADAPT system,will be verified. The verification is necessary for the project group to be able to show the customer/orderer that the requirements are fulfilled.

Many requirements are trivially easy to verify, for example the ones stating that an analysis of some kind shall be performed. In those cases it is easy to check (e.g. in the Technical Documentation) and see if the analysis in question has been performed. Such easily verifiable requirements are omitted from the Test Plan.

2 System modeling

This section handles ways to verify the requirements that is connected to the requirements made on modelling.

2.1 Requirement 6

This requirement can be verified by feeding the model as an XML-document into the diagnosis program created by the project group. If the program thereafter produces diagnoses as intended, we have verified the requirement. The easiest way to check that the program produces diagnoses as intended is to use the training data sets**?** and look at what diagnoses are produced.

3 Diagnostic algorithm

This section handles ways to verify the requirements that is made on the diagnosis algorithm and its ability to detect and isolate faults.

3.1 Requirement 8

To verify this requirement an analysis of which single faults that are detectable has to be made first. Provided that analysis, run the diagnosis program for all the training data sets**?** and check that all detectable single faults among that data are detected by the diagnosis program.

3.2 Requirement 9

Verification of this requirement is done analogous to the verification of requirement 8, except that instead of single faults we are dealing with double faults, and therefore scenarios containg double faults is used instead.

3.3 Requirement 10

Verification of this requirement is done analogous to the verification of requirement 8, except that instead of single faults we are dealing with multiple faults of higher order

than two, and therefore scenarios containg multiple faults of order two and higher, is used instead.

3.4 Requirement 12

To verify this requirement an analysis of which single faults that are isolable has to be made first. Provided that analysis, run the diagnosis program for all the training data sets**?** and check that all isolable single faults among that data are isolated by the diagnosis program.

3.5 Requirement 13

Verification of this requirement is done analogous to the verification of requirement 12, except that instead of single faults we are dealing with double faults.

3.6 Requirement 14

Verification of this requirement is done analogous to the verification of requirement 8, except that instead of single faults we are dealing with multiple faults of higher order than two.

4 Software

This section handles ways to verify that the requirements that is made on the software is fullfilled.

4.1 Requirement 18

This is a basic requirement that needs to be fulfilled in order for the program to work, i.e. to be able to interact with DxC so it can get scenario data, record diagnoses, evaluate the result etc. Therefore this requirement is easily verified by checking for example that scenario results are generated.

4.2 Requirement 19

Verification of this requirement is done by looking at the source code and checking that correct datatypes are used.

4.3 Requirement 21

The startup time is measured on a regular computer, e.g. any of those in the Project area of the Division of Vehicular Systems (swe. *FS Projektrum*).

4.4 Requirement 24

This requirement means that the program shall be modifiable so that removal of a sensor is handled, i.e. all test variables using the sensor in question are removed from the program

and the program shall still be able to produce a diagnosis. Verification is done by removing a sensor (deleting the sensor data), modifying the program accordingly, and running the program to see if a diagnosis is produced.

4.5 Requirement 25

This requirement is tested by feeding the diagnosis program training data sets which has been trimmed at the end.