

ERTMS/ETCS – Class 1

Test Sequence Generation : Methodology and Rules

REF : SUBSET-076-4-1

ISSUE : Version 1.0.2

DATE : 25/02/2009

Company	Technical Approval	Management approval
ALSTOM		
ANSALDO SIGNAL		
BOMBARDIER		
INVENSYS RAIL		
SIEMENS		
THALES		

1. MODIFICATION HISTORY

Issue Number Date	Section Number	Modification / Description	Author
0.0.1 24 October 2000	All	Creation. Document Subset- 076-4 Generation of Test trips	A. Göhler
0.1.0 6 April 2001	-	Delivery edition. Document Subset- 076-4 Generation of Test trips	Ch. Frerichs
0.1.1 5 July 2002	6.2	Addition of points 7 and 8 to the list of principles. Document Subset- 076-4 Generation of Test trips	Daniel Molina
2.2.0 30 July 2002	-	Editorial changes and comments received from UNISIG partners. Document Subset- 076-4 Generation of Test trips	Daniel Molina
0.0.1 17 October 2002	All	First draft. Change of the name of the document Subset-076-4-1: Test Sequences Generation : Methodology and Rules.	DLR (Meyer zu Hoerste)
0.0.2 18 October 2002	All	Modification after discussion Subset-076-4-1: Test Sequences Generation : Methodology and Rules.	DLR (Jaschke)
0.0.3 19 October 2002	All	Modification after discussion Subset-076-4-1: Test Sequences Generation : Methodology and Rules.	DLR (Meyer zu Hoerste)
0.0.4 30 October 2002	Chapter 4 – 6	Modification after crosscheck with ISV (Günther) and discussion Subset-076-4-1: Test Sequences Generation : Methodology and Rules.	DLR (Jaschke)

	Chapter 7	Created Subset-076-4-1: Test Sequences Generation : Methodology and Rules.	Alcatel (Goss)
0.0.5 1 November 2002	All	Included comments from other partners Subset-076-4-1: Test Sequences Generation : Methodology and Rules.	DLR (Jaschke)
	Section 4.5	Created Subset-076-4-1: Test Sequences Generation : Methodology and Rules.	DLR (Meyer zu Hoerste, Schwartz, Jaschke)
1.0.0 30 April 2003	All	Edition for delivery including both the Test sequences Methodology and the Rules used to write down the Test Sequences Subset-076-4-1: Test Sequences Generation : Methodology and Rules.	CEDEX Jorge Iglesias DLR Stefanie Schwartz Michael Meyer zu Hörste
1.0.1 14 April 2008	All	Alignment to SRS 2.3.0. Ready for delivery	Oscar Rebollo
1.0.2 25 February 2009	3, 4, 9.5	Editorial changes	Oscar Rebollo

2. TABLE OF CONTENTS

1. MODIFICATION HISTORY.....	2
2. TABLE OF CONTENTS.....	4
3. INTRODUCTION.....	6
4. PRINCIPLES.....	7
5. PURPOSE OF TEST SEQUENCES	8
6. TEST SEQUENCES – GIST	10
6.1 Introduction	10
6.2 Selection of Test Cases for Frames.....	10
6.3 Graphical Explanation	11
6.4 Types of frames	13
6.5 States	13
6.5.1 All States.....	13
6.5.2 Essential States	14
6.6 Transitions	15
6.6.1 Identity and Level transitions.....	15
6.6.2 Merged Transitions	15
6.7 Conclusion – Number of test frames.....	15
7. METHOD OF CONSTRUCTION OF TEST SEQUENCES	17
7.1 Concatenation of Test Cases to Test Sub-sequences	17
7.2 Grouping of test sub-sequences into test frames.....	17
7.3 Concatenation of Test Sub-sequences to Test Sequences.....	17
8. TEST SEQUENCE GENERATION.....	18
8.1 Graphical Representation	18
8.2 Constraints.....	19
8.3 Objectives	19
8.4 The Mathematical Algorithm	19
8.4.1 Assumptions	20
8.4.2 Advantages.....	20
8.4.3 Disadvantages	21
9. DOCUMENTATION OF TEST SEQUENCES.....	22
9.1 Definition of a Start or an End State.....	22
9.2 The Documentation of a Test Sub-sequence.....	22
9.3 Test Sub-sequence Template	22
9.4 The Documentation of a Test Frame.....	23

9.5	The Documentation of a Test Sequence.....	23
9.6	Terms and Definitions (Glossary).....	23
9.6.1	Feature	23
9.6.2	State	24
9.6.3	Test Case	24
9.6.4	Test Sub-sequence.....	24
9.6.5	Test Frame	24
9.6.6	Test Sequence.....	24
10.	SELECTION OF TEST CASES FOR FRAMES.....	26
10.1	First Selection (Start State, Transition, End State)	26
10.2	EXAMPLE: Test Cases of Frame LSTM SE → L1 FS	26
10.3	Definition of global and local parameters.....	27
10.3.1	Global Parameters	27
10.3.2	Local parameters	27
10.4	EXAMPLE: Test Case Sequence of Frame LSTM SE → L1 FS	28
11.	RULES TO CREATE TEST SUB-SEQUENCES.....	29
11.1	Introduction.....	29
11.2	Feature Assignment	29
11.3	Use of generic TC's.....	29
11.4	TSS generation	30
11.5	Advice	31
11.6	Start of Mission.....	31
11.7	Level Transitions	32
11.8	Removal of Transitions.....	32
11.9	Procedure to modify Test Sub-sequences.....	32
12.	RULES FOR THE CREATION OF TEST SEQUENCES WITH TSW.....	33
12.1	Introduction.....	33
12.2	General rules.....	33
12.3	TSW display	34

3. INTRODUCTION

This document is written in the frame of the “Test Specifications WG”. It describes how Test Sequences are generated. This document is valid in the scope of the test specification as defined in the test plan (Subset 076-0, version 2.3.1, section 2.2.1).

This document is part of the ERTMS/ETCS documentation. The general procedure that has to be applied to produce test specifications is defined in the Test Plan SUBSET-076-0.

The Test Sequence creation process is hereby described in detail. Test Sequences are based on the Feature List SUBSET-076-5-1 and in the Test Cases SUBSET-076-5-2, their main goal consists of providing a set of Laboratory trips, the so-called Test Sequences, in order to check that an On-Board system is compliant with the SUBSET-026, issue 2.3.0

Besides, this document shows the Methodology applied by the Test Specification WG to write down the set of Test Sequences contained in the document SUBSET-076-6-3 , as well as the set of rules defined by this WG to create Test Sequences.

The set of Test Sequences included in the SUBSET-076-6-3, issue 2.3.0 , covers the list of requirements related to ERTMS/ETCS On-Board system for Level 0/1/2, including RIU and Euroloop. Level 3 will be included in future releases.

4. PRINCIPLES

The principles at issue are as follows:

- 4.1.1.1 Test cases have been written down according to the procedure defined in the document “Methodology of testing”, SUBSET-076-3. As test cases are used to build up different test trips and the output of one Test Case is the input for the next Test Case, a common format to describe Test Cases has had to be used.
- 4.1.1.2 One Test Sequence consists of at least one Test Case.
- 4.1.1.3 There is no upper limit for the number of Test Cases.
- 4.1.1.4 One Test Case can only follow another one if the output data of the first test case fit to the input data of the second one.
- 4.1.1.5 Test trips must be generated in a way that they can be run in the test environment, which should be similar to the one defined in the SUBSET-094-0 “UNISIG Functional Requirements for an on-board Reference Test Facility”.
- 4.1.1.6 The complete set of test cases covers only testable requirements of the UNISIG SRS. In order to make sure that all these testable requirements of the UNISIG SRS are tested, all test cases have to be used at least once inside one of the Test Sequences.
- 4.1.1.7 The agreement among the WG states that each Test Case has to be checked at least once in one of the applicable mode-level combinations. Therefore, it is not necessary to check the Test Cases in more than one mode-level combination.
- 4.1.1.8 Pursuant to this agreement, the Test Sequence set, essential to ensure the Interoperability of both On-board and Trackside sub-systems, is clearly defined, that is to say, these Test Sequences should gather all Test Cases and it is enough to pass each Test Case in at least one mode-level combination.
- 4.1.1.9 The multiple use of the same Test Cases shall be minimised by means of a convenient design of every Test Sequence.
- 4.1.1.10 The interoperability tests can be carried out not only with prototypes but also with commercial equipment. For the latter, there is no possibility whatsoever to enter the black box. This implies that the equipment to be tested must be in an event state for the start and the end of the Test Sequence. The start of a Test Sequence could be reached thus by means of standard operations.

5. PURPOSE OF TEST SEQUENCES

The purpose of building up Test sequences is justified in the following issues:

- 5.1.1.1 Test Cases do not allow to be tested individually. It is necessary that the system reaches the initial state. The only way to attain this is by using the system functionality (Start of Mission, information reception by balise or radio etc...). As this functionality is included in other TCs, some sort of concatenation is needed. The way of performing this concatenation has to be carefully analysed, since some requirements of the SRS have to be tested jointly or by following a precise order. Therefore, the Test Sequences should not be created arbitrarily.
- 5.1.1.2 The Test Cases being grouped into Test Sequences provide a UNIFORM interpretation of the SRS, standardising the performed Tests:
 - 5.1.1.2.1 Train Speed Profile and Messages sent to the train (balise, loop or radio) have to be clearly defined while writing down a Test Sequence. A different approach to a Test Sequence could mean a different interpretation of a specific TC. This would imply that non-STANDARD Tests were possible.
 - 5.1.1.2.2 During the process of the Test Sequence creation, some ordinary mistakes have been discovered in the TCs and corrected accordingly. If, contrary to this process, each laboratory had written its own Test Sequences, some mistakes in the TCs (already known) have been equally corrected for TCs in order to be implemented into a Lab. The difference is that, in the latter way, testing according to the TSI would not have been just possible.
- 5.1.1.3 The Test Sequences constitute a very RELIABLE basis for testing, allowing for the black box principle. The Test Cases contain the tables of internal states, not available at any standard interface. However, the Test Sequences only contain FFFIS or FIS (TIU, SSS and DMI as defined in the SUBSET-094).
- 5.1.1.4 Test Sequences define the minimal functionality blocks (per Level) to be covered by each ERTMS/ETCS on-board equipment, in this way, full Interoperability is actually guaranteed. The following blocks could be tested separately in the Laboratory:
 - Block 1: L 0-1-2.
 - Block 2: L 0-1 .
 - Block 3: L 0-1-2 +RIU and Loop
 - Block 4: L 0-1-2+RIU+Loop+STM.

Test Sequences such as defined in the SUBSET-076-6-3, issue 2.3.0 , The other three blocks will be covered by future extensions of the Test Sequences.

5.1.1.5 If the set of Test Sequences had not been written down, each Notify Body would need to check on their own:

- whether any particular set of Test sequences performed in a Lab covered all the TCs,
- whether they were correctly concatenated (according to SRS),
- whether they guaranteed all the requirement fulfilment.

Consequently, if the Test Sequences had not been written down, it would have been really difficult to standardise the NoBo's certification process.

5.1.1.6 The set of Test Cases contained in the SUBSET-076-6-3 ensures that all the L0,1-and-2 Test Cases are tested at least once and therefore that all the SRS requirements related to these three levels are also tested. This is compliant with the requirement written in the document "Methodology of testing SUBSET-076-3".

6. TEST SEQUENCES – GIST

6.1 Introduction

6.1.1.1 As it has been explained in the previous section, Test Cases have to be tested and gathered into a Sequence. In order to do this, the Test Cases have to be concatenated. The main goal of the Methodology showed in this document is to define the way of concatenating Test Cases.

6.1.1.2 The Methodology has been focused on a definition of a set of internal states covering all possible combinations of Level and Mode. A transition between two internal states is called a Test Frame. In the following paragraphs are described all Frames considered in the Methodology. Once the Frames had been defined, the WG has written down a set of Test Sub-sequences inside each Frame, by concatenating Test Cases assigned or related to the Mode and/or Level Frame.

6.1.1.3 The complete set of Test Sub-sequences has to cover all Test Cases at least once so as to fulfil the requirement of testing all Test Cases. The final step of the Method consists of concatenating the sub-sequences into the final Test Sequences. This process has been carried out by means of a Mathematical Algorithm which ensures that all sub-sequences are used at least once as well as that the number of used sub-sequences are minimised. Therefore, at the end of this process, all Test Cases would have been tested at least once and also all the possible Mode/Level transitions according to SRS 2.3.0.

6.1.1.4 To sum up:

- A Test Sequence is a set of concatenated Test Sub-sequences.
- A Test Sub-sequence is a set of concatenated Test Cases.
- Some Test Sub-sequences are grouped (not concatenated) into a Test Frame.
- A Test Frame defines all possible transitions between two states by a set of Test Sub-sequences starting at the initial state and ending at the final one.
- When both the initial and the final state of a Frame are the same, this Frame is called an Identity.
- The complete set of Test Sequences has to cover all Test Sub-sequences.
- The complete set of Test Sub-sequences has to cover all Test Cases.

6.2 Selection of Test Cases for Frames

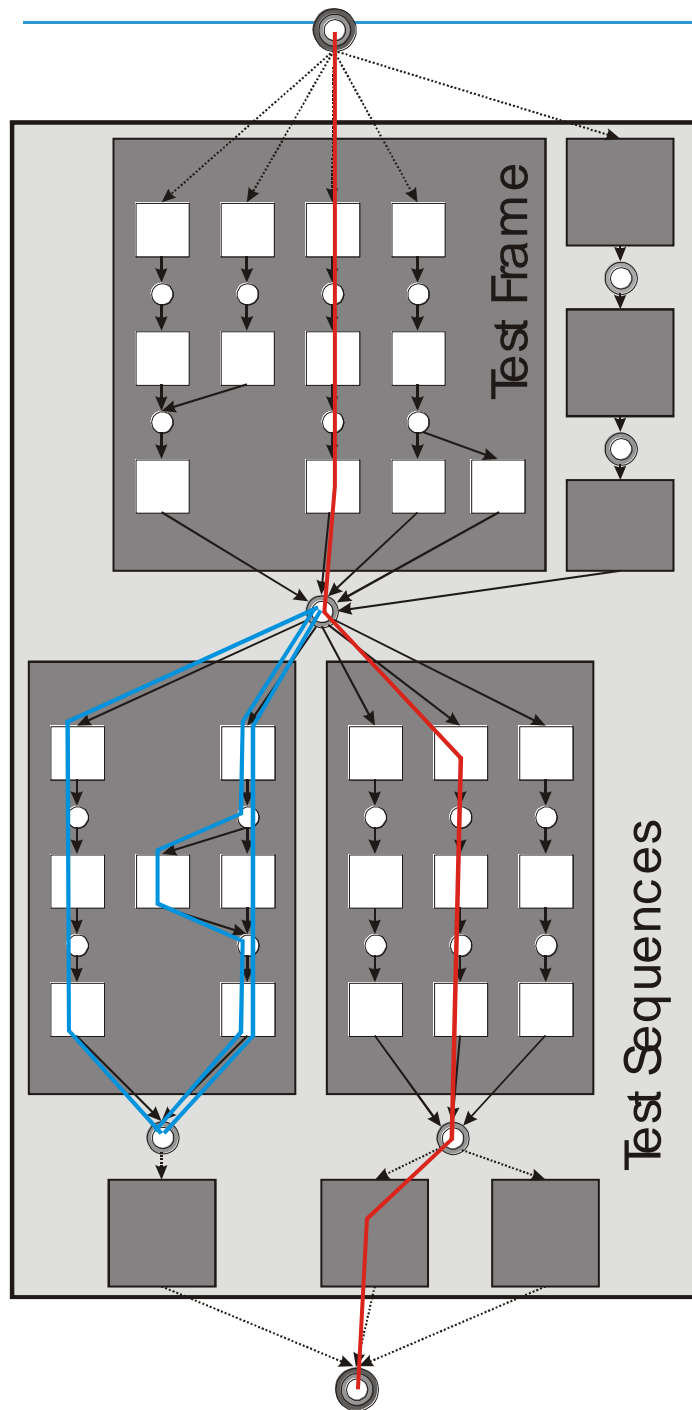
6.2.1.1 This document also describes the way to select Test Cases to be assigned to a specific Frame. The selection criteria have to guarantee that all relevant TC's for this

specific frame are selected. Moreover, a proposal is made for the definition of global and local parameters to further reduce the resulting set. The tester is intended to be able to select, with the help of global and local parameters, for a frame exactly those test cases which are appropriate for a specific test situation.

6.3 Graphical Explanation

The internal states are represented by all possible mode/level combinations. Each transition from one mode/level combination to another one (where either the mode or the level, or, in some exceptional cases, both the mode and the level may change) is handled by a test frame. There are also frames to handle identities which mean no level or mode transition. Each frame consists of one or more test Sub-sequences that enables the concatenation of one or more test cases.

Figure 1 below shows the gist of concatenating Test Cases to Test Sub-sequences by gathering Test Sub-sequences into Frames and finally concatenating Sub-sequences from different Frames into Test Sequences. To illustrate this process, three Sub-sequences and one sequence are shown in fig. 1 .



Legend:

- ⊙ Sequence Start/End State (NP)
- Test Frame Start/End State
- Test Case Start/End State

- Test Case
- Test Subsequence
- Test Frame
- Test Sequences
- Test Sequence

Fig. 1: General Approach

6.4 Types of frames

There exist four types of test frames:

1. Test frames which contain a mode transition but no level transition.
2. Test frames which contain a level transition but no mode transition.
3. Test frames which contain both a level transition and a mode transition.
4. Test frames which contain neither a level transition nor a mode transition.

Most of test frames containing a level transition as well as a mode transition have been split up into two different test frames: one that contains a level transition and one with a mode transition.

The next step has been to detect all necessary start/end states for the frames.

Constraints:

- The transition conditions have to be taken into account.
- Each mode transition must be present in at least one frame.
- Different mode transitions with identical transition conditions could be represented in one frame, only if the lowest level covers all transitions.

6.5 States

6.5.1 All States

The total number of start-and-end states could be detected in the following way:

Using the SRS 2.3.0 Chapters 4.6.2 and 4.6.3., a description can be derived from as shown in the table “all states” (see excel sheet States for Test Sequences, SUBSET-076-4-2). This excel sheet shows all start and end states.

- Number of start states: 55
- Number of end states: 55

6.5.2 Essential States

Some transitions are performed in different levels. Since the common knowledge is that all transitions are tested in the lowest level, it is possible to merge some levels according to one mode, when the transitions (UNISIG SRS 2.3.0 Chapter 4.6.2) are equal. The page “Reduced states” of the excel sheet States for Test Sequences shows all essential merged start-and-end states.

- Full number of states: 55
- Number of equivalent states: 20
- Number of essential states: 35

The equivalent states are showed in the table below (see also States for Test Sequences , “Equivalent states”):

State	Equivalent states	State	Equivalent states
FS : L1	-	SB : L2	SB : L3
FS : L2	-	SB : STM	-
FS : L3	-	SE:STM	-
IS : L0	-	SF : L0	-
IS : L1	IS : L2; IS : L3	SF : L1	SF : L2; SF : L3
IS : STM	-	SF : STM	-
NL : L0	-	SH : L0	-
NL : L1	NL : L2; NL : L3	SH : L1	-
NL : STM	-	SH : L2	SH : L3
NP : L0	-	SL : L0	SL : L1; SL : L2; SL : L3; SL : STM
NP : L1	NP : L2; NP : L3	SN : STM	-
NP : STM	-	SR : L1	-
OS : L1	-	SR : L2	SR : L3
OS : L2	OS : L3	TR : L1	-
PT : L1	-	TR : L2	TR : L3
PT : L2	PT : L3	UN : 0	-
RV : L1	RV : L2; RV : L3		
SB : L0	-		
SB : L1	-		

Table 1: Equivalent States

6.6 Transitions

6.6.1 Identity and Level transitions

The basis for the transitions according to SRS 2.3.0 Chapters 4.6.2 are the mode transitions. However, there have also been considered features and test cases where no mode transition but a level transition exists, or neither a mode nor a level transition is made – the so-called “identity”-. Considering the grouped mode/level combinations showed in the table “reduced states” of the excel sheet States for Test Sequences , a reduced number of identities and level transitions has been obtained. This number appears in the table 2 below. The number of identities results to be identical to the number of essential states.

6.6.2 Merged Transitions

Some transitions, e.g. from FS L2/3 to SH L2/3, are reached by more than one transition. For instance, the transitions 6, 50 and 51 lead from FS L2/3 to SH L2/3. The transition from OS L2/3 to SH L2/3 is also achieved by the transitions 6, 50 and 51. Therefore, transition combinations could be merged into one “transition identifier”.

The excel sheet States for Test Sequences “transition ID” shows all necessary concatenated start-and-end states and the merged transitions. A table with the transition ID’s and the rest of transitions according to the SRS is also included there.

Frames for transitions made up with inseparable mode and level transitions are considered within the mode transitions.

6.7 Conclusion – Number of test frames

The varied concatenations among level transitions and mode transitions lead to the following necessary number of frames:

The sheet “Test Frames” in the Excel file “States for Test Sequences” shows two different types of test frames: “Relevant test frames” are needed to test the functionality of the SRS, “Currently non- used test frames” may be necessary to reach a specific state but they just cover a functionality that is already covered by a mandatory test frame.

Relevant test frame: Equal mode transitions within different levels could be grouped into one transition within the base level (Level 1), called relevant test frames.

Currently non-used test frames: Identical mode transitions are grouped together but, in case the concatenation of the relevant test frames leads to a “dead end”, these currently non-used frames must be used.

	All	Equivalent	Relevant	Currently non-used
<i>States</i>	55	18	37	0
Identity Frames	55	18	23	14
Mode Transition Frames	349	98	97	154
Level Transition Frames	90	4	42	44
Sum of Frames	494	120	162	212

Table 2: Number of States and Frames

7. METHOD OF CONSTRUCTION OF TEST SEQUENCES

7.1 Concatenation of Test Cases to Test Sub-sequences

The states described in the excel file “States for Test Sequences”, SUBSET-076-4-2 are used to define the start and end conditions of the test sub-sequences. The test sub-sequences have been developed by concatenation of test cases.

7.2 Grouping of test sub-sequences into test frames

The same states defined in the excel file “States for Test Sequences”, SUBSET-076-4-2 are used to define the start and end conditions of the test frames. All possible sub-sequences leading from a particular start state to a particular end one are grouped within a test frame.

7.3 Concatenation of Test Sub-sequences to Test Sequences

Those states defined in the excel file “States for Test Sequences”, SUBSET-076-4-2, which are used to define the start and end conditions of test frames, are the connecting points between two test frames. Test sequences have been designed by concatenation of test sub-sequences. By definition, the start state and end state of a test sequence is No Power (NP).

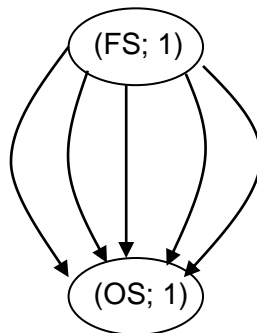
8. TEST SEQUENCE GENERATION

8.1 Graphical Representation

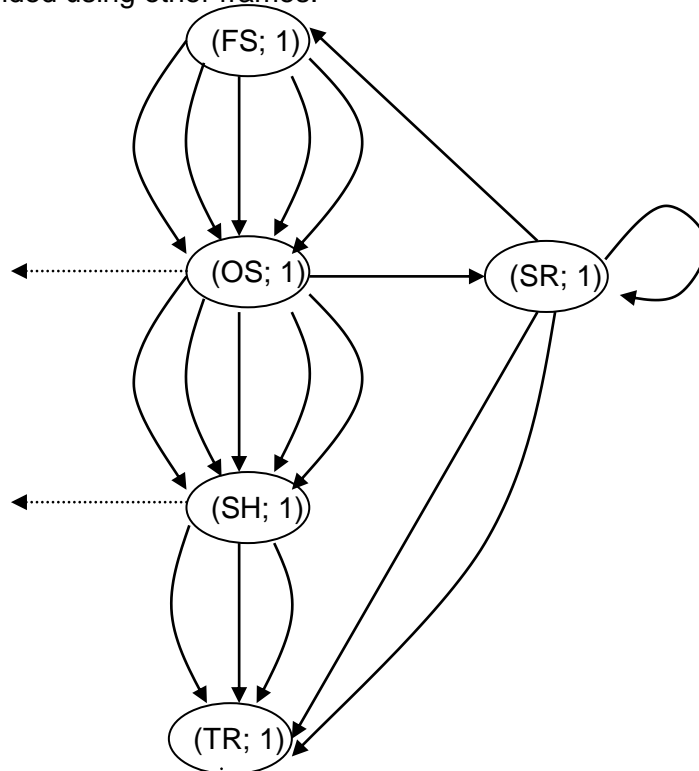
Every frame has a well-defined start and end state, e.g. frame 006: start state (FS; 1), end state (OS; 1). Every sub-sequence belonging to this frame leads from the start state to the end one.

Each state is represented through a vertex and each subsequence through an arch.

Example: Frame 006 from (FS; 1) to (OS; 1) has 5 sub-sequences, so the graphical representation has 5 arches from vertex (FS; 1) to vertex (OS; 1).



this graph may be extended using other frames:



This subsequently leads to a complex graph, hard to handle manually, showing one arch for each subsequence that has been written down.

8.2 Constraints

We have to test each test case at least once. The sub-sequences will cover all test cases.

=> If we test every subsequence at least once, we test every test case once or more.

Every test sequence we create should start in NP and end in NP. A test sequence is a path through the graph that starts and ends in NP.

8.3 Objectives

The main objectives are:

- To test every subsequence at least once,
- To start and end in NP,
- To test as few sub-sequences as possible more than once (therefore, to minimise the test effort and avoid unnecessary work).

These objectives imply a well-known problem from the graph theory, the Chinese Postman Problem.

8.4 The Mathematical Algorithm

The used mathematical algorithm (The Chinese Postman Problem) is explained as follows: The Chinese Postman has to deliver letters to people. To do this, he walks down the streets where people live. He has to walk down every street to deliver all his letters but he does not like to walk streets down twice, if he does not need to do so.

In this case, all streets the Chinese Postman has to cover are one-way streets. As a consequence, we have to solve out a special Chinese Postman Problem: the Chinese Postman Problem for directed graphs.

There is a known algorithm that finds an optimal tour for the Chinese Postman. This algorithm is available for free and has been implemented in various programming languages.

If we apply the algorithm to our graph of states and sub-sequence, we get one long test sequence that covers all sub-sequences and minimises the amount of testing.

Through these test sequences, we reach the state (NP; 1, 2, 3) several times (at least 11 times as this is the amount of Start of Mission frames). Every time we reach (NP; 1, 2, 3), we make a cut. In such a way, we split up the long way of the Chinese Postman into several parts which are our test sequences.

8.4.1 Assumptions

- Test sub-sequences are independent, that is to say, there are no dependencies among them: neither “followed by” nor “preceded by”.

8.4.2 Advantages

- All sub-sequences are used. No additional attention needs to be paid to this point.
- The amount of testing is minimised.
- The algorithm is ready. No tool has to be developed.
- Test sequences are generated before the final version of the frames is delivered. The algorithm only requires precise information on the start states, the end states and the number of sub-sequences of each frame.
- It allows focusing on special sub-sequences or on those needless to test very often and rarely used, which are the ones starting or ending in NP and those that lead to IS.
- The sub-sequences form a directed graph. Such a graph might not be strongly connected¹. This means that either some sub-sequences cannot be reached from NP, or NP cannot be reached from some sequences (dead ends). Therefore, to be able to build up a sequence that contains such sub-sequences, new frames have to be written.
However, In order to build up a set of sequences to cover all sub-sequences, the graph must be strongly connected. The Chinese Postman Algorithm checks whether the graph is strongly connected or not. Thus we can determine additionally needed frames at an early stage.

¹ A directed graph is called “strongly connected” if for each two vertices x and y there is a directed path from x to y.

8.4.3 Disadvantages

- This approach has been developed for a laboratory test, further operational constraints have not been considered.
- In principle, the algorithm behaves quite randomly. The kind and number of test sequences cannot be predicted, only estimated though they can be forced by means of assigning different weight to different sub-sequences.
- All test sequences are generated as a whole. This task cannot be split up into different stages, if the aim is to minimise the test effort.

9. DOCUMENTATION OF TEST SEQUENCES

9.1 Definition of a Start or an End State

The definition of a state implies:

- The current or related mode.
- The current or related level.

9.2 The Documentation of a Test Sub-sequence

The documentation of a test sub-sequence must contain:

- The unique identifier of the test subsequence.
- The start state and the end state according to “states_v07.xls”, table “start & end states”.
- The identifiers of used test cases and their chronology.
- Constraints.

9.3 Test Sub-sequence Template

Sub-sequence 1

COMMENTS
<i>Short description, start / end state, constraints, comments</i>

Step	Feature		Test Case			Mode / Level
	#	Name	#	Test Case of Feature	Uses	
1	X	<i>name of feature X</i>	Y	<i>name of test case Y of feature X</i>	<i>Saa-Fbbb-TCccc</i> OR <i>empty</i>	<i>start mode,</i> <i>start level =></i> <i>end mode,</i> <i>end level</i> OR <i>mode, level</i>

2		
---	-----	-----	-----	-----	--	--

9.4 The Documentation of a Test Frame

The documentation of a test frame must consist of:

- The unique identifier of the test frame.
- The start state and the end state according to “states_v07.xls”.
- The identifiers of the used test sub-sequences. If more than one are possible, all sub-sequences have to be listed.
- Constraints.
- Date and author.

9.5 The Documentation of a Test Sequence

The documentation of a test sequence must include:

- The unique identifier of the test sequence.
- A graphical representation of the test sequence (concatenation from test frames).
- The identifiers of used test frames and test sub-sequences and their chronology.
- Constraints.
- Date and author.

The start state and the end state is No Power and train at standstill. Hence, the start state and the end state do not have to be defined separately for each test sequence.

9.6 Terms and Definitions (Glossary)

9.6.1 Feature

A feature is the definition of one functionality of the SRS (Subset 26). A feature is tested by one or more test cases. The identifier is the feature number (F No.).

9.6.2 State

A state consists of:

- A mode
- A level

Three classes of states are used to assemble test sequences, test frames, test sub-sequences and test cases:

1. Test sequence state: the state which a test sequence starts in or ends up in. There is only one state: No Power (NP).
2. Significant states: test frames and test sub-sequences start or end in these states.
3. State: test cases start or end in these states.

9.6.3 Test Case

A test case is one test of a feature. To test a feature completely, either one test case or several test cases may be necessary. The identifier is the feature number together with the test case number (F No. TC No.).

9.6.4 Test Sub-sequence

A test sub-sequence is a chronology of test cases in a test frame, which leads from the start state to the end state. The identification is the test frame and test sub-sequence number (F No. SubS No.).

9.6.5 Test Frame

A test frame is a gathering of test sub-sequences, which lead from one particular start state to one particular end state. A test frame may consist of several test sub-sequences. The identification is the test frame number (TF No.).

9.6.6 Test Sequence

A test sequence is a chronology of test sub-sequences. The identification is the test sequence number (TS No.), the test frame numbers (TF No.) and the test sub-sequence numbers (SubS No.).

10. SELECTION OF TEST CASES FOR FRAMES

10.1 First Selection (Start State, Transition, End State)

10.1.1.1 In the first step, all relevant test cases have been selected according to:

- A) Start State
- B) Transition (Level, Mode)
- C) End State

10.2 EXAMPLE: Test Cases of Frame LSTM SE → L1 FS

State/Level	Selection Criteria	# Features	# TCs
LSTM	Level = STM and Mode = European	3	4
LSTM Supervision	Level = STM and Mode = European	3	4
LSTM SHOW	Level = STM and Mode = European	3	3
LSTM Driver	Level = STM and Mode = European	2	2
LSTM / L1	Transition LSTM → L1	5	9
L1	Mode = FS	16	32
L1 Transmission	Mode = FS	17	61
L1 Supervision	Mode = FS	31	50
L1 SHOW	Mode = FS	16	29
L1 Driver	Mode = FS	1	1
L1 Brake	No brake intervention or brake intervention is inhibited/ permitted	6	6
L1 Distance	Mode = FS	7	21
L1 Loop, Balise, Balise Infill, Radio Infill	Mode = FS	61	132
Sum		167	350

Note: Detailed list of TC's see LSTM_L1_TC's.xls, folder first selection

- 10.2.1.1 The resulting set guarantees, generally speaking, that all defined test cases are involved, independent of specific sequences within the frame.
- 10.2.1.2 This set of test cases is selected according to level and mode and may be further selected on a finer granularity according to global parameters of the sequence and characteristics of track sections.
- 10.2.1.3 Global characteristics may be available technologies and data transmission, which may be determined by a generic trackside or predefined, input which is to be received by the on-board system.

10.2.1.4 For each frame, the relevant test cases can be selected individually or dynamically, depending on the specification of the track.

10.3 Definition of global and local parameters

10.3.1 Global Parameters

10.3.1.1 Parameters which refer to more than one frame of a sequence are considered as global.

Examples for global parameters:

- Profile data, valid for more than one frame
- Available trackside and data transmission technologies,
- Available train/OBU equipment, ETCS version

10.3.1.2 For instance, transmission technology applied in L1 = Balise

State/Level	Selection Criteria	# Features	# TC's
L1	Transmission = Balise and Mode = FS	100	191

Note: For detailed list of TC's, see LSTM_L1_TC's.xls, folder second selection

10.3.1.3 The number of relevant TC's has been reduced to 191.

10.3.2 Local parameters

10.3.2.1 Local parameters are the ones referring only to one frame.

Examples for local parameters:

- Data input (driver input such as acknowledgement, override EOA, etc..)
- Profile data, valid for a sole frame
- Type of supervision (release speed, danger points, override EOA)

10.3.2.2 For instance, static speed profile and route suitability data are transmitted and supervised.

State/Level	Selection Criteria	# Features	# TC's
L1	Transmission of SSP and route suitability data Supervision of SSP and route suitability data	91	170

Note: For detailed list of TC's, see LSTM_L1_TC's.xls, folder third selection

10.3.2.3 The number of relevant TC's has been reduced to 170.

10.4 EXAMPLE: Test Case Sequence of Frame LSTM SE → L1 FS

10.4.1.1 Definition and navigation into sequences are possible with the help of dependencies such as preceded by, followed by and “uses” – relationships which have been specified among the test cases.

10.4.1.2 Global parameters of the test trip:
- Transmission technology = Balise

10.4.1.3 Local parameters of the frame:
- Transmission of SSP

- L1 speed supervision

10.4.1.4 All relevant test cases have been selected according to the Selection criteria. The following sequence has been defined from these test cases.

TC Nr.	Level/Mode	Description	Followed by
262-2	LSTM	Initiation of Start of Mission through Driver	TC1: 561 TC2: 577 or 579 TC3, TC4: 552 TC5: 555 or 547 TC6: 557 or 548 TC7: 549
577-1	LSTM	Driver selects “Start” and ETCS level is STM and SE mode is required for the STM (Transition [57])	262-2
549-1	LSTM	Train Data is valid and MA and track description is available and mode profile requires no specific mode (Transition [10])	
37-1	LSTM	Transmission of Static Speed Profile via Balise group	For level STM -> 1 one of the TC of feature 485
485-1	LSTM	Announcement with Indication of level transition L STM/L1 from track device other than RBC	
486-1	LSTM	Order to switch immediately to Level 1 at level border (Level transition LSTM / L1)	566 or 633 or 550 or 554 or 560 or 425 or 460
550-1	L1	ETCS Level switches to 1, 2 or 3 MA and track description is available and mode profile requires no specific mode (Transition [25])	
595-1	L1 FS	Handling of supervision data at entering full supervision mode	

10.4.1.5 The hierarchy of the test cases defined by “preceded by”-and-“followed by” relations has not been completely defined yet.

11. RULES TO CREATE TEST SUB-SEQUENCES

11.1 Introduction

- 11.1.1.1 Section 12 brings together the rules which have been used for the creation of the Test Sub-sequences (TSS) in the Test Frames (TF).
- 11.1.1.2 This document has been used as a work document for internal use while creating the Test Sub-sequences. The main target at distributing it is to be used for future creation of additional Test sub-sequences.

11.2 Feature Assignment

- 11.2.1.1 Every On-board Test Case (TC) in each Feature must be tested at least once.
- 11.2.1.2 TC's related to a mode transition must be assigned to the appropriate mode transition frame.
- 11.2.1.3 TC's related to a level transition must be assigned to the appropriate level transition frame.
- 11.2.1.4 The indication of mode and level must be tested generally in the mode and level transition frames or Start of Mission, where they appear newly.
- 11.2.1.5 If some piece of information has to be received on-board, such information must be transferred by a suitable message in the suitable TC's.
- 11.2.1.6 Starting conditions of the first TC and End conditions of the last TC of a sub-sequence must be described shortly in the comment box so as to find compatible sub-sequences for concatenation.

11.3 Use of generic TC's

- 11.3.1.1 A list of generic TC's has been provided. They have been assigned to special transitions or identities.
- 11.3.1.2 In Levels 2 and 3 is the Radio Connection supposed to be established. Provided that the radio connection is lost, this has to be sorted out by the use of a suitable TC's.
- 11.3.1.3 Indications of time or position should be tested just once in an identity frame or in the start of mission frame.

- 11.3.1.4 If different transmission TC's are possible, they must be used within the different TSS's of a Frame (e.g. position report from a single balise or two balises).

11.4 TSS generation

- 11.4.1.1 If the USE_F#nnn is not included in the TC, no comment should be made to the TC but it must be firstly ensured that the TSS is complete and correct.

- 11.4.1.2 If the USE_F#nnn is included in the TC (or a comment meaning the same), the procedure should be the following:

A – Firstly, include the Feature and TC used in the TSS, using a new column:

S#aa-F#bbb-TC#ccc

S#dd-F#eee-TC#fff

The S#aa means: aa is the step number of the sequence of test inside the TC where the USE_F#bbb is included (see diagram).

The TC number shall be selected for each TSS and could be different in different TSS.

If the original Feature includes the identification of the TC to be used, do not modify it.

B – Secondly, make sure that the Feature and TC already included in one step of the TSS are not repeated in the following step (if not necessary).

C - Once the Features are concatenated in the TSS and one step inside the TC's is repeated, include as follows in the new column:

NS#aa-F#bbb-TC#ccc <reason> ,

OR

NS#aa-F#bbb <reason>

(without "#" of course)

Use this possibility if it is clearly repeated. It makes the Test Sequence revision easier.

- 11.4.1.3 If other USE_F#bbb are found within a previously "called" Test Case, the secondary "USE" commands must be written in brackets:

S#aa-F#bbb-TC#ccc (S#pp-F#qqq-TC#rrr (S#ss-F#ttt-TC#uuu ...))

S#aa-F#bbb-TC#ccc (NS#pp-F#qqq-TC#rrr)

S#aa-F#bbb-TC#ccc (NS#pp-F#qqq)

S#aa-F#bbb-TC#ccc (S#pp-F#qqq-TC#rrr (NS#ss-F#ttt-TC#uuu))

S#aa-F#bbb-TC#ccc (S#pp-F#qqq-TC#rrr (NS#ss-F#ttt))



Note: Neither “<reason>” nor “Remove_Step#aa” commands must be used inside these secondary “USE” commands.

11.4.1.4 If any step of the test case currently in use is not applied, add the following comment:

Remove_Step#aa <reason>

(i.e. if there is a comment such as “only in L2 / L3”, being actually in L1)

11.5 Advice

11.5.1.1 It should be avoided to write all possible combinations of different TC’s within a Frame. The main objective is to cover all the TC’s of a unique Frame and the usually operational situations within it.

11.5.1.2 Four additional Frames for “Limit of Authority at Level Border” have been added (FS L 2,3 -> TR L1; OS L2,3 -> TR L1; FS L1 -> TR L2,3; OS L1 -> TR L2,3), whose numbers have been assigned by DLR.

11.5.1.3 Sub-sequences must be written in the following order:

- Mode transition
- Indication of new mode
- Inform trackside
- Handling of data

11.5.1.4 More information, such as “standstill or not”, “radio communication established”, etc., should be added in the comments.

11.6 Start of Mission

11.6.1.1 Start at NP and end in one of the “grey boxes” (SRS - Start of Mission Procedure).

11.6.1.2 All boxes have to be covered.

11.6.1.3 The mode transitions inside the SoM Procedure are included.

11.6.1.4 Used Frames for the mode transitions inside the SoM are recorded.

11.7 Level Transitions

11.7.1.1 Transitions L2 and L3 have been set apart. As a result, the additional frames have been assigned to each company that is doing the “other” level transition. Frame numbers have been assigned by DLR.

11.8 Removal of Transitions

11.8.1.1 Features that remove transitions are included into identity frames.

11.9 Procedure to modify Test Sub-sequences

11.9.1.1 In case a new TSS needs to be generated inside a Frame, it must be added as a new one. If an old TSS is going to be replaced, the author must ensure that the new one covers at least the same TC's (but for TC's included in another TSS). This TSS is to be collected by the WPL to be assigned a new TSS number. The reason why this new TSS is created must be written by the author in the modification history of the Frame and in the comments of the TSS.

11.9.1.2 A document containing a list of all these new TSS has been created by the WPL.

12. RULES FOR THE CREATION OF TEST SEQUENCES WITH TSW

12.1 Introduction

- 12.1.1.1 Section 13 gathers the rules used for the creation of Test Sequences (TS). These are general rules to harmonise the way of writing down the Test sequences.
- 12.1.1.2 Test sequences have been created by means of the Test Sequence Wizard Tool. This Tool allows an easy creation of the Sequence by collecting the Sub-sequences, concatenating them, generating the Speed Profile, ordering the steps of the complete Sequence and bringing in the balise and radio messages.
- 12.1.1.3 The Test sequence wizard generates an output which is a Word file including all the necessary Test sequence information. It also creates a bitmap showing the Speed profile defined for each Sequence.
- 12.1.1.4 The TSW also contains all the information related to the Sequence in an electronic format which is an Access file. Therefore, an automatic translation of all data from the TSW to a Test Bench may be possible, avoiding thus all sorts of errors coming from a manual implementation of this set of Test Sequences into Laboratory Tools.

12.2 General rules

- 12.2.1.1 Whenever an event has to be recorded in the JRU, it has to be tested.
- 12.2.1.2 Whenever an event has to be displayed in the DMI, it has to be tested.
- 12.2.1.3 “Handling of supervision data” features must be tested after each single transition.
- 12.2.1.4 “Permanent” checking can be avoided by testing these events only once in the particular Sequence.
- 12.2.1.5 Cyclically tested events, such as speed recording, can be tested on only in the particular Sequence.
- 12.2.1.6 We have to add package 255 in all balise telegrams.
- 12.2.1.7 During SoM, the on-board does not receive an order to contact RBC. The on-board shall establish the communication session automatically if possible.
- 12.2.1.8 Each partner is free to propose an alternative Chinese Postman file for the Sequences in order to improve the Sequence design. Avoid removing TSSs that cause TCs to be uncovered.

- 12.2.1.9 When switching to Trip from SH is not possible to perform a level transition.
- 12.2.1.10 When performing a Level transition to L0 and STM being in Trip mode, the acknowledgement of the level transition must be performed before the acknowledgement of Trip mode.
- 12.2.1.11 In sequences containing “Rejection of L1, 2, 3 MA” features when the on-board is in L0 (without having received level transition announcement to L1, 2, 3), we must transmit both packets 12, 21 and 27 together.
- 12.2.1.12 The packet 80 for OS/SH modes shall be sent together with a movement authority. The movement authority is composed of the packet 12, together with the track description, packets 21 and 27. If the track description is not available, the MA shall not be used.
- 12.2.1.13 Balises within the same BG shall be located according to Subset-040 (4.1.1.1 and 4.1.1.2) specs.
- 12.2.1.14 Distance / time in TSW steps window are only mandatory for communication points but optional for other devices such as DMI, JRU, etc. These values shall be mandatory for lab testing.
- 12.2.1.15 The Speed profile must describe real EVC + driver speed behaviour. It is optional to fill in the comments box in TSW related to the speed driver that should be followed for testing the Sequence in lab (by adding “Speed: x km/h”). These values shall be mandatory for lab testing.

12.3 TSW display

- 12.3.1.1 RED is used when a step has been removed (manually or due to TSS importation)
- 12.3.1.2 GREEN is used when a step has been moved
- 12.3.1.3 GREY is used when “NS” command appears.