EVC Architecture Analysis

Model Description and Interfaces

**Summary:**

**Company:** Alstom  
**Authors:** Christian GIRAUD  
**Reference:** openETCS - WP3  
**Index:** <index>  
**Date 1st issue:** 20/02/2015

Documentation for Modelling

**Contents:**

[Chapter 1: Overall 3](#_Toc435104773)

[Architecture 3](#_Toc435104774)

[Breakdown 4](#_Toc435104775)

[Database Objectives and Principles 6](#_Toc435104776)

[Basic Positioning 6](#_Toc435104777)

[Positioning with “Linking” 6](#_Toc435104778)

[Database 7](#_Toc435104779)

[Chapter 2: Speed Control Process 9](#_Toc435104780)

[Supervision Modes 9](#_Toc435104781)

[Speed Control Model in Excel 9](#_Toc435104782)

[Chapter 3: Scope of WP3 15](#_Toc435104783)

[Annex 1: MRSP and Asafe 17](#_Toc435104784)

[Annex 2: Real Time Aspect 19](#_Toc435104785)

[Annex 3: Modes & Levels 20](#_Toc435104786)

[Annex 4: Internal Data 24](#_Toc435104787)

[Annex 5: Speed and Distance Monitoring, Model in SCADE Suite 26](#_Toc435104788)

[Préconisations 30](#_Toc435104789)

[Annex 6: « steps » of random size 30](#_Toc435104790)

[1. Method of sorting 30](#_Toc435104791)

[2. Interpolation 31](#_Toc435104792)

# Chapter 1: Overall

## Architecture

Remind architecture yet described in « Papyrus »:

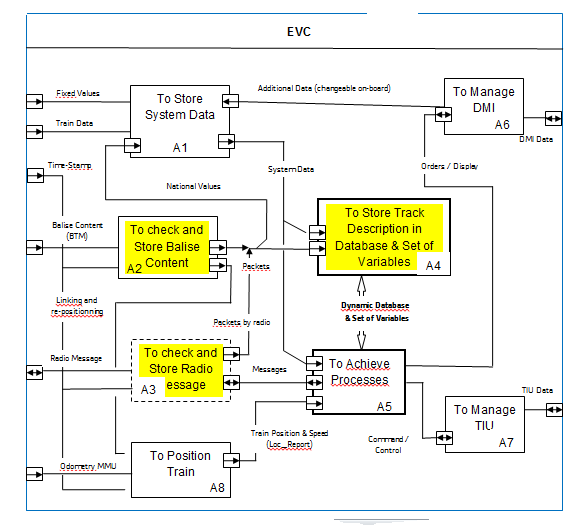


Diagram [A0]

Comments:

A1: To Store System Data:

This box is a general purpose which gathers all parameters such as:

“Train Data” that are defined during the on-board data preparation,

“Fixed Values” that are generic,

“National Values” that come from transmission (or trackside) and

“Additional Data” that can be defined by the driver.

A2: To Check and Store Balise Data:

This box deals with:

All data received by BTM and transmitted to EVC,

Computation of exact train position through MMU and Time stamping.

A3: To Check and Store Radio Messages:

All data transmitted by Radio and transmitted to EVC,

Computation of exact train position through MMU and Time stamping.

A4: To Store Track Description in Database:

Database is managed through elementary functions:

To store packets into one or several events,

To withdraw,

To keep in order.

Description is given later on.

A5: To Achieve Processes:

The list of processes is given later on.

A6: To Manage DMI:

All processes to display to driver or receive from.

A7: To Manage TIU:

All processes to order to train or receive from.

A8: To Position Train:

To acquire data from odometer,

To position train in database (Max and Min Front End, Antenna).

“Speed and Distance Monitoring” is a part of box A5:

## Breakdown

The final breakdown structure of IBD “second level” could be established as hereafter, by distinguishing mandatory and secondary functions.

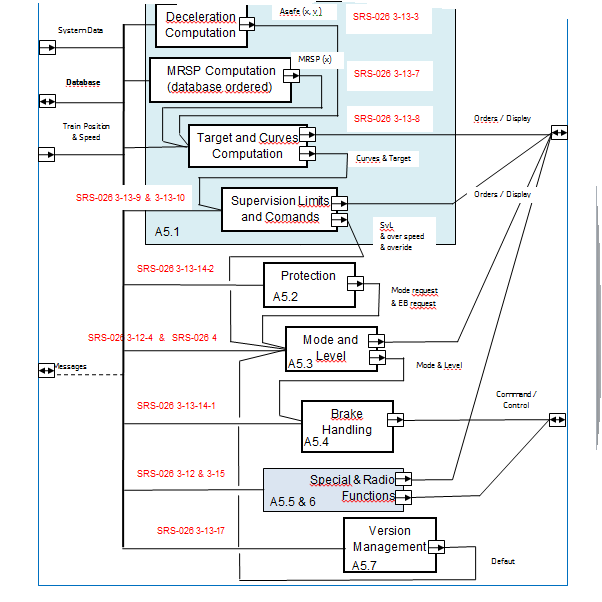


Diagram [A5]

A5.1: Speed and Distance Monitoring:

* A5.1.1: Deceleration( x, v ), Build-up Time, Gradients, Rotating Mass computation,
* A5.1.2: MRSP(x ) computation (Most Restrictive Speed Profile),
* A5.1.3: Targets and Curves computation, MRDT (Most Restrictive Displayed Target),
* A5.1.4: Supervision Limits computation (SvL, Ceiling, Target, Release),
* A5.1.5: Commands of Speed and Distance Monitoring.

A5.2: Protection:

* A5.2.1: Emergency Stop (UES, CES, revocation, inhibition)
* A5.2.2: Track ahead Free,
* A5.2.3: MA Shorten,
* A5.2.4: Roll Away Protection,
* A5.2.5: Reverse Movement Protection,
* A5.2.6: Standstill Supervision.

A5.3: Mode and Level Monitoring:

* A5.3.1: Level Handling,
* A5.3.2: Mode Handling.

A5.4: Brake Command Handling.

## Database Objectives and Principles

* To permit communication between “Technical Functions” (boxes A2, A3, A8) and “Operational Functions” (box A5).
* To decompress trackside data in order to permit computation related to train position and speed.
* To have a common “Coordinate System” for all functions.
* To have a common “Coordinate System” between trackside and train-borne.

Positioning is based on an “Absolute Counter” of odometer which is composed of three values in reason of multiple inaccurate sources of odometer sensors:

* Nominal value: given by current nominal counter of odometer.
* Maximum value: given by maximum counter of odometer.
* Minimum value: given by minimum counter of odometer.

## Basic Positioning

All these values are corrected by an estimation of run distance during elapsed time between odometer and EVC. This estimation is achieved through time stamp and odometer speed estimation. Maximum and Minimum value must be within range of +/-5% of Nominal value between two BG. This correction is: Delta\_T \* V\_est.

All these values are set to zero at power up and are never reset until power cut. Meanwhile, when overpassing one new LRBG, maximum and minimum are reset to nominal value (+/- balise position inaccuracy), by keeping in memory the 3 counters current value for use later on.

* Nominal LRBG value: nominal value when overpassing last LRBG.
* Maximum LRBG value: maximum value when overpassing last LRBG.
* Minimum LRBG value: minimum value when overpassing last LRBG.

When no linking is used, only the checking of 5% is taken into account and the 32 bits odometer counter is used to position any event in database. Correlation between 32bits counter and address is achieved modulo ”N”. Example:

* Quantum = 5 m,
* Distance since power-up = 1000 km = 10exp6 m,
* Address modulo 1K = (10exp6 / 5) modulo [ 1024] = 320 = 140H,
* Coverage Maximum = [ 1024] \* 5 = 5,12 km

## Positioning with “Linking”

When overpassing a linked balise group with linking available, this BG has a double position:

* Odometer counter when overpassing the BG,
* Anticipated counter through previous BG counter position increased by “D\_Link”.

A “Algorithm” is involved to choice between both, each counter having its own window value.

Basically, anticipated counter should be chosen when D-Link is free of error while odometer counter is disturbed by sensor inaccuracy. The result is so-called “Estimated Position” and window limits are so-called Max-Safe-Front-End and Min-Safe-Front-End.

As long as linking is available, the estimated position is the sum-up of D-link value.

When linking is no longer available, odometer counter is again the odometer reference.

**Chronogram about reading of BG:**

Cn-1

Cn+1

Cn

D\_link(n)

D\_link(n+1)

BGn-1

BGn

BGn+1

Train Positioning

When passing over BGn-1:

* Odometer counter is Cn-1
* Data attached to BGn-1 are positioned at [Cn-1 + x] because no linking is given for BGn-1
* BGn will be expected at [Cn-1 + D\_link(n)]

When passing over BGn:

* Odometer counter is Cn, BGn and Cn become the reference:
* BGn is seen in advance related to odometry counter = Delta
* Delta = [Cn-1 + D\_link(n) - Cn]
* Data attached to BGn are positioned at [Cn + x + Delta]
* BGn+1 will be expected at [Cn + D\_link(n+1)]

## Database

**DataBase Build Up**

The build-up is achieved in 3 phases:

Phase 1: Event are extracted from BG telegram or Radio message, two cases must be envisioned:

* The event being the first LRBG: no special caution,
* The event being a new LRBG: old event have to be swept following type (for instance, a new MA requires cancellation of SSP, grade profile, but TSR is (are) saved) .

In any case, the odometer counter permits to calculate the BG address modulo the database size.

Phase 2: From Train Position up to SvL, calculate MRSP, A\_Gradient and K\_Adh from event positioned in phase 1, by using all System Data as necessary.

Phase 3: Calculate Curves.

**Database Use:**

As long as no new BG or new radio message are to be taken into account, the database is frozen

The curves calculated in phase 3 are used through “Train Position” and “Train Speed”. See “X0” on diagram, V0 being the triggering emergency brake of EBI curve.

The monitoring mode (CSM, TSM and RSM) and status (normal, indication, permit, warning, over speeding, intervention, SB, EB) are calculated by using curves and train speed and position.

**Database Synopsis:**

Position Type DOT. Data1 Data2 MRSP AGrd K\_Adh Target EB\_ G\_ K\_ Corr\_ENER

Energy Computation:

EBD, EBI, SBD,

SBI1, SBI2

DMI curves: W, P, I

See Events Definition

Preliminary Computation

Follows odometry

(step = 5 m )

Train Running Direction

Data from Packets

(used for preliminary computation)

Database Architecture

# Chapter 2: Speed Control Process

## Supervision Modes

Three modes are involved:

* Ceiling Speed Monitoring: this is the phase when speed is stable (CSM).
* CSM is characterized by one berthing speed and one allowance of overspeeding.
* Berthing Speed is so-called the Permitted Speed.
* Berthing Speed plus Allowance defines the EBI (Emergency Brake Intervention).
* No EBD is defined.
* Target Speed Monitoring: this is the phase when speed is reduced (TSM).
* TSM is characterized by one target speed or reduction speed and a deceleration.
* Both target and deceleration define the EBD (Emergency Brake Deceleration).
* Deceleration is a function of MRSP.
* A model is involved to take into account all response time related to breaking.
* Modelisation can be approached through two ways:
* for a given position “X0”, to calculate the related speed “V0” for all curves (EBD, IBD, SBI, W, P). This is the preferred solution.
* for a given speed “V0”, to calculate the related position “X0” for all curves. This is the SRS description.
* Release Speed Monitoring: this is the phase when approaching a stop (RSM).
* RSM is characterized by a release of EBI curve which is replaced by an override control at the EOA.
* This release is defined by a so-called release speed whose value depends on distance from EOA to DP.

## Speed Control Model in Excel

The use or Excel is only to specify the curves computation in order to obtain graphic curves attached to signal status.

Modelling requires the following variables:

* Position: location per quantum of 10 m.
* Choice of decreasing position is done,
* The EOA is supposed to be at position 0 of the turning buffer,
* Any other position can be done for EOA.
* Vc: EBD curve is starting from a target defined by a couple {X1, V1}.
* X1 = position of target,
* V1 = speed reduction of MRSP,
* EOA is initial target,
* Curve is computed by step of 10 m with: V(i+1) = SQRT [ V(i) ² + (2\*Asafe(i) \*10)]
* Scanning is achieved from farther target up to train position,
* Vc’: EBD curve for IBD is extended EBD curve in ceiling speed monitoring,
* Target with speed above EBD are ignored.
* Asafe: deceleration in m/s² for emergency braking curve.
* Asafe = Aeb + Agrad /Alpha.
* Aeb = emergency brake deceleration value,
* Aeb = function of MRSP speed and Position (5 values maximum),
* Agrad = acceleration / deceleration value due to slope and gravity,
* Agrad = p \* 9.81 with p>0 in uphill and p<0 in downhill, taking in account worse case over train length,
* Alpha = coefficient of turning mass, acting as amplifier on mass in movement,
* Alpha = 1.xx with 4% < xx < 15%.
* MRSP: compile SSP, ASP, Veoa, Vrelease, TSR, in m/s,
* dV: delta speed for MRSP allowance,
* MRSP+dV: includes dV to define EBI in ceiling speed monitoring.
* VT: target speed = last speed reduction when reading from EOA toward train position,
* VT is taken into account when VT = V0 or VT < V0.
* TSM: target speed monitoring,
* TSM = true when Vc’ < MRSP + dV,
* Dbec: See drawing :
* DT: T1 + T2 ,
* T1 is Ttraction, or duration of Aest1,
* T2 is Tberem, or duration of Aest2,
* DX: used to calculate Dbec: V0 \* DT + DX,
* DV: is equal to: (T1\*Aest1) + (T2\*Aest2).
* Second Degree Equation: a\*x² + b\*x + c,
* a = 1,
* b: coefficient “b”,
* c: coefficient “c”,
* D: discriminant = b² - ( 4\*a\*c )
* V0: EBI speed,
* Solution of Second Degree Equation,
* V0 = [ SQRT( D) – b ] / 2,
* Vbec: EBI speed + DV.

**D**iagram Distance versus Speed Control

We define a diagram which give the speed related to each curve for a given train position.

For EBD, EBI, SBI, W and P, we get a speed value starting from train position.

V0 = f( X0) is 5 functions (EBD, EBI, SBI, W and P).

The comparison with Vest permits to define 6 possible status:

Trip: Vest > EBD or (EOA overpassed) or (Timer elapsed),

EB: Vest > EBI or ( (Vest > Vrelease) and RSM),

SB: Vest > SB

W: Vest > W

P: Vest > P

Null: Vest < P

Dbec

Delta V

Vbec = f( Dbec, Asafe)

V0 = Vb – Delta\_V

Xb

X0

X1

V1

Vc = f( X0, X1, V1, Asafe)

**Computation of EBD curve and EBI curve through V0 = Function( X0 )**

Vc² = V1² + 2\*(X1-X0)\*Asafe = **f(X0, X1, V1)**

Delta\_V = (Aest1 \* T1) + (Aest2 \* T2) = **DV**

Dbec = ½ (Aest1\*T1² + Aest2\*T2²) + (Aest1\*T1\*T2) + V0\*(T1 + T2) = **DX + (V0 \* DT)**

Vbec² = Vc² – 2\*Dbec\*Asafe) ( 1st expression based on V1)

= Vc² - 2\*Asafe\*(DX + V0\*DT) = V1² + 2\*Asafe\*(X1-X0-DX-(V0\*DT))

Vbec² = (V0 + DV)² = V0² + 2\*V0\*DV + DV² ( 2nd expression based on V0)

**V0 solution = elimination of Vbec:**

**V1² + 2\*Asafe\*(X1-X0) – 2\*Asafe\*(DX + V0\*DT) = V0² + 2\*V0\*DV + DV²**

**V0² + 2\*V0\*(DV + Asafe\*DT) + 2\*Asafe\*DX + DV² - V1² - 2\*Asafe\*(X1-X0) = 0**

a = 1 b = 2\*(DV + Asafe\*DT) c = 2\*Asafe\*(DX+X0-X1) + DV² - V1²

D = ( b² - 4\*a\*c) / 2 a

V0 = [ - b + SQRT( D ) ] / 2 => **V0 = - (DV + Asafe\*DT) + ½ SQRT(D**)

**Verification Invariant: DT = 0 must make V0 = Vc.**

DT = 0 => D = 4 \* [ V1²/2 + Asafe\*(X0-X1) ]

DT = 0 => V0 = ½ SQRT(D) = SQRT( V1² + 2\*Asafe\*(X0-X1) ) = Vc

« C »

« B »

« 0 »

« 1 »

« A »

Computation of V0=f(X0)

**Computation of EBD curve and EBI curve through X0 = Function( V0 )**

**V0² + 2\*V0\*(DV + Asafe\*DT) + 2\*Asafe\*DX + DV² - Vc² = 0**

Vc² = V1² + 2\*(X1-X0)\*Asafe

V0² + 2\*V0\*(DV + Asafe\*DT) + 2\*Asafe\*DX + DV² = V1² + 2\*Asafe\*(X1-X0)

V0² - V1² + 2\*V0\*(DV + Asafe\*DT) + 2\*Asafe\*DX + DV² = 2\*Asafe\*(X1-X0)

V0² - V1² + 2\*V0\*DV + DV² + 2\*V0\*Asafe\*DT) + 2\*Asafe\*DX = 2\*Asafe\*(X1-X0)

(V0+DV)² - V1² + 2\*Asafe\*(V0\*DT + DX) = 2\*Asafe\*(X1-X0)

(X1-X0) = [ (V0+DV)² - V1² ] / (2\*Asafe) + (V0\*DT + DX)

with V0 + DV = Vbec and V0\*DT + DX = Dbec

(X1-X0) = [ Vbec² - V1² ] / (2\*Asafe) + Dbec

**X0 = X1 - [ Vbec² - V1² ] / (2\*Asafe) - Dbec**

**Then:**

**X0 = X1 - [ ( V0 + DV )² - V1² ] / (2\*Asafe) - [ DX + ( V0 \* DT) ]**

Problem: X0 = F(V0° is not a function because a same “V0” can provide several “X0”.

Computation of X0

# Chapter 3: Scope of WP3

We need to come back to architecture datagram at the beginning of the present document.

The scope of “group 1” is limited to boxes A51 and A52:

* A5.1: Speed and Distance Monitoring, based on excel file n°400 and n°401 ( mode OS / FS):
* A5.1.1: Deceleration( x, v ), Build-up Time, Gradients, Rotating Mass computation:
  + - Computation of Asafe(x, v) with train characteristics, grade profile, adhesion factor and rotating mass coefficient,
    - Computation of build-up Time (variables DX, DT, DV).
* A5.1.2: MRSP(x ) computation (Most Restrictive Speed Profile):
  + - Computation of MRSP(x) and dV with Speed Profile (SSP and ASP), TSR, Max Train Speed and Max Signaling Speed.
* A5.1.3: Targets and Curves computation, MRDT (Most Restrictive Displayed Target):
  + - Computation of EBD curve based on MRSP(x) and Asafe(x, v),
    - Computation of Target (variables VT, XT),
    - Computation of Monitoring Mode (variable TSM),
    - v being equal to MRSP(x) + dV,
    - Scan starting from EOA or DP or OL.
* A5.1.4: Supervision Limits computation (SvL, Ceiling, Target, Release):
  + - Computation of SBI, W, P and I curves.
* A5.1.5: Commands of Speed and Distance Monitoring:
  + - Taking in consideration the train position (X0), defines the speed limit for EB, SB, W, P and I,
    - Taking in consideration the train speed, defines the status and outputs (Trip, COP, SB, W, P, I).
* A5.2: Protection (mode OS, FS, LS):
* A5.2.1: Emergency Stop (UES, CES, Revocation) by radio message:
  + - UES (Unconditional Emergency Stop) is radio message n°16,
      * Is referenced by an Id and addressed to one train, without position,
      * Always taken into account by onboard,
      * Can be revocated by message n° 18,
      * Acknowledgement by message n° 147.
    - CES (Conditional Emergency Stop) is radio message n°15,
      * Is referenced by an Id and addressed to one train, with position,
      * Is taken in account if max-safe-front-end has not overpassed its position,
      * Can be revocated by message n° 18,
      * Acknowledgement by message n° 147.
* A5.2.2: Track ahead Free by radio message:
  + - Track Ahead Free Request is sent to train by message n° 34,
    - Track Ahead Free Granted is sent back to wayside by message n° 149.
* A5.2.3: MA Shorten by radio message:
  + - MA Shorten is requested to train by message n° 9,
    - a new position of EOA is provided by trackside,
    - Request to Shorten MA is granted by on-board with message n° 137,
    - Request to Shorten MA is rejected by on-board with message n° 138.
* A5.2.4: Roll Away Protection:
  + - Movement in reverse direction is limited between 2 LRBG.
* A5.2.5: Reverse Movement Protection:
  + - Reverse Movement are monitored by packets n°138 and n°139.

A5.2.6: Standstill Supervision:

* + - Standstill is provided by odometer,
    - is used to exit Trip Mode.

# Annex 1: MRSP and Asafe

Computation of MRSP( x) and Asafe( x, v) by starting from packet inputs.

Computation is based on one matrix of 2 dimensions:

First dimension = Position:

Odometer Counter Modulo S, S being size of turning buffer,

Quantum = 1 m, 5000 positions, coverage = 5 km.

Second dimension = Data related to MRSP and Asafe computation:

Data inputs: SSP, QF, Grad, ASP, Veoa, TSR, Adhesion, Aeb0, Aeb1, Aeb2, Aeb3,

Data outputs: MRSP(x), Asafe0(x), Asafe1(x), Asafe2(x), Asafe3( x).

LRBG

Pk B

EOA

Pk E

LRBG

Pk\_A

**LRBG** SSP QF ASP Veoa TSR **MRSP** Grd Adh Aeb0, 1, 2, 3 **Asafe0, 1, 2, 3**

DOT

Other

Pk X

**LRBG** SSP QF ASP Veoa TSR **MRSP** Grd Adh Aeb0, 1, 2, 3 **Asafe0, 1, 2, 3**

**LRBG** SSP QF ASP Veoa TSR **MRSP** Grd Adh Aeb0, 1, 2, 3 **Asafe0, 1, 2, 3**

Linking Distance A-B

Train Position

Turning Buffer Architecture

The train is somewhere between Pk\_B (Last Relevant Balise Group) and Pk\_E (Pk related to EOA).

**MRSP( x) computation:**

All received packets are distributed over the matrix, considering that linking distances are free of error and initial speed of a section is final speed of the previous section.

SSP values according position (D\_STATIC & V\_STATIC), and according train length (QF),

ASP values,

Veoa value,

TSR values according position, TSR length and train length (option).

LRBG value is defined when overpassing linked BG:

SSP values are coming from packet n°27:

ASP values are coming from packet n° 51:

Veoa value is coming from packet n°12 or n°15:

TSR values are coming from packet n°65:

**Asafe( x) computation:**

All received packets are distributed over the matrix:

Grd values, uphill/downhill, compensated by train length,

Adh values, permit to define four steps Aeb0, Aeb1, Aeb2, Aeb3.

**Conclusion:**

Such an operation on compacted data, in order to make data easy for computing, requires to address data at digit level.

As result, we should prefer to compute these vectors through a dedicated “C. Code” program.

For this reason, **train length** is not taken into account within the SCADE Model in computation of MRSP and Asafe. Should it be reasonable as this variable is not declared as “safe”?

Therefore, data are designed following “worst case” of train length.

# Annex 2: Real Time Aspect

The real-time aspect is taken into account through 3 levels of software:

Level 0: period 2 ms or 10 ms, to execute urgent actions and timers.

Level 1: period 100 ms up to 500 ms, to execute cyclic processes (speed and distance monitoring).

Level 2: aperiodic operation of several seconds (MRSP, Asafe and Curves computation in “background”).

New LRBG

New Data Received by radio

New Data transferred in turning buffer

Duration for computation and transfert

Duration for transmission

Real Time Period

Real Time Architecture

**Choice of quantum:**

There is no need to compute the database with a constant value of quantum.

Therefore, we shall choose the quantum value in accordance with the target distance TD.

TD > 2000 m , then Q = 50m , ( 400 records for TD = 20000 m),

TD > 200 m , then Q = 5 m , ( 400 records for TD = 2000 m),

TD < 200 m , then Q = 0,5 m, ( 400 records for TD = 200 m).

# Annex 3: Modes & Levels

**Introduction**

« Modes and Levels » should be considered as the major function of box [A5: To Achieve Processes].

At each process is attached one procedure.

Most of procedures are attached to one Mode and can be chosen by driver.

Levels depend on on-board and trackside equipment and cannot be chosen by driver.

List of Modes and Procedures:

We distinguish “Basic Procedures” and “Secondary Procedures”.

**Basic Procedures**

|  |  |
| --- | --- |
| Mode | Procedure |
| SB | Stand-by,  Procedures “Start of Mission” & “End of Mission”. |
| SR | Staff Responsible,  Procedure “Override”. |
| FS | Full Supervision,  Procedures UES, CES, MA deployment. |
| OS | On Sight,  Procedure On-Sight. |
| TR / PT | Trip, Post Trip,  Procedure Trip. |
| Level  Transition | NTC, 0, 1, 2, 3. |

**Secondary Procedures**

|  |  |
| --- | --- |
| Mode / status | Procedure |
| Standstill | Change of Orientation |
| Standstill | Reversing |
| Standstill | Splitting |
| Standstill | Joining |
| Standstill | Change of Driver |
| Level 1, 2, 3 | Track Conditions |
| Level 2 / 3 | Handover RBC |
| Any | Limited Supervision |
| Any | LX Management |
| Any | Shunting |

**State of Machine**

Following basic procedures, we can describe the following State of Machine with 20 transitions.

SR

FS

OS

TR

PT

(MA+..level1) 31,

(MA+..level2/3) 32

40 (OS enforcement)

31, 32

37(Push Button Override)

12, 16, 17, 18, 20, 41, 65,66, 69

12, 16, 17, 18, 20, 41, 65,66, 69

18, 20, 36, 42, 43, 54, 65

31, > FS

15, > OS

8, 37, > SR

8 ( Driver Ack SR ),

37 (Push Button Override)

SB

(Desk Close) 28

37(Push Button Override)

15 ( Driver Ack OS ),

40 ( OS Enforcement)

(MA+..level1) 31,

(MA+..level2/3) 32

7 ( Driver Ack TR ),

& (Standstill) & (level1,2,3)

Modes, State of Machine

This « State of Machine » is a subset of SRS-026, the number of transitions is limited by 21.

|  |  |
| --- | --- |
| 7 | (Driver acknowledge Trip) AND (Standstill = True) AND (Level = 1, 2 or 3) |
| 8 | (Driver acknowledge proposed SR) |
| 12 | (Level = 1) AND (Min Safe Antenna has overpassed EOA) |
| 15 | (Driver acknowledge proposed OS) |
| 16 | (Level = 2 or 3) AND (Min Safe Front End has overpassed EOA) |
| 17 | (Linking Reaction) |
| 18 | (Balise Trip Order) AND (Override not Activated) |
| 20 | (UES is active) |
| 28 | (Desks closing) |
| 31 | (MA+SSP+GRD are available) AND (Mode Profile empty) AND (Level = 2 or 3) |
| 32 | (MA+SSP+GRD are available) AND (Mode Profile empty) AND (Level = 1) AND (Trip Order from Balise) |
| 36 | (Balise Unexpected in SR list) AND (Override not Activated) |
| 37 | (Override Activated) AND (No Override Overspeeding) |
| 40 | (Mode Profile enforces OS) AND (Max Safe Front End is inside OS area) |
| 41 | (T\_NVCONTACT is elapsed) AND (Reaction is required) |
| 42 | (SR distance is overpassed) AND (Override is not active) |
| 43 | (MA Timer has elapsed) AND (Override is not active) |
| 54 | (Stop in SR has been received) AND (Balise is not expected) AND (Override is no longer active) |
| 65 | (Version number given by balise is higher than expected) |
| 66 | (Linked Balise is passed in unexpected direction) |
| 69 | (Estimated Front End is in rear of SSP or Grd start location) |

Modes, transition

**NB: transition 43 is supposed to take in consideration all elapsed timers attached to MA and managed by “Speed and Distance” monitoring.**

Complete list of Transition (in yellow, transition toward Trip Mode):

|  |  |  |
| --- | --- | --- |
| Name | Origin | Use |
| On-board Isolated | TIU | Graph Mode |
| One Desk Open | TIU | Graph Mode |
| Go Sleeping | TIU | Graph Mode |
| On-board Powered | TIU | Graph Mode |
| Standstill | Odometer | Graph Mode |
| Level = 2,3 | Level Graph | Graph Mode |
| Level = 1 | Level Graph | Graph Mode |
| Level = 0, NTC | Level Graph | Graph Mode |
| SH required from Driver | MMI |  |
| SH granted from RBC | Radio Reception |  |
| Trip Acknowledge from Driver | MMI |  |
| SR offered from EVC | SR Graph |  |
| SR Acknowledge from Driver | MMI |  |
| Train Data Valid | Train Data |  |
| MA-SSP-GRD valid | Database |  |
| Mode Profile empty | Database |  |
| EOA overpassed by MinSafeAntenna | Speed & Distance Monitoring |  |
| Fail Safe Failure | EVC |  |
| OS required from Trackside | Radio Reception |  |
| OS Acknowledge from Driver | MMI |  |
| EOA overpassed by MinSafeFrontEnd | Speed & Distance Monitoring |  |
| Linking Reaction | Train Positionning |  |
| Trip from Timer Elapsed | Speed & Distance Monitoring |  |
| Override Active | Override Graph |  |
| UES accepted | Speed & Distance Monitoring |  |
| SH end required by Driver | MMI |  |
| End of SH | SH Graph |  |
| Passive Shunting Receive | TIU? |  |
| No Power of EVC | TIU |  |
| External Trip | Balise |  |
| Mode Profile = OS at MaxSafeFrontEnd | Database |  |
| Train Speed <= Max Override Speed | Speed & Distance Monitoring |  |
| Non Leading Signal is received | TIU |  |
| Non Leading is selected by Driver | MMI |  |
| Stop Order in SH out of override | Balise / Radio |  |
| Stop Order in SR out of override | Balise / Radio |  |
| BG in SH out of list SH | Balise |  |
| BG in SR out of list SR | Balise |  |
| Linked Balise in Wrong Direction | Balise |  |
| Wrong Version of Balise | Balise |  |
| Estimated Front End is in rear of SSP or Grd start location | Balise / Radio |  |

Modes, elementary transition

# Annex 4: Internal Data

This annex describes data by using only “Integer”.

SCADE model is using only “Floating Point” in order to simplify.

**Data Definition**

Distance (D):

32 bits

31 8 | 7 0

bit 0 = 1/256 meter (m)

bit 8 = 1 m

Sign = bit 31

Range = [-8 388 608.. +8 388 607,99 ]

Velocity (V):

16 bits

15 8 | 7 0

bit 0 = 1/256 meter per second (m/s)

bit 7 = 1 m/s

Sign = bit 15

Range = [-128.. + 127,99 ]

Acceleration (A):

16 bits

15 8 | 7 0

bit 0 = 1/256 meter per second square (m/s²)

bit 8 = 1 m/s²

Sign = bit 15

Range = [-128.. + 127,99 ]

Energy (E):

32 bits

bit 0 = 1/256² meter square per second square (m²/s²)

bit 16 = 1 m²/s²

31 16 | 15 0

Sign = bit 31

Range = [ - 32 768.. + 32 767,99 ]

Energy corresponds to “Speed Square” and would be used only to avoid the use of “Square Root”.

Time:

16 bits

bit 0 = 10 ms

Sign = bit 15

Gradient :

16 bits

bit 0 = 1/1000

Sign = bit 15

**Computation**

Multiplication ( V over 16 bits ) \* ( V over 16 bits ) / 2 = E over 32 bits

Multiplication ( D over 32 bits ) \* ( A over 16 bits ) / 256 = E over 32 bits

Multiplication ( A over 16 bits ) \* ( Time over 16 bits) / 100 = V over 16 bits

Multiplication ( Gradient over 16 bits ) \* 981 / 100 000 = A over 16 bits

# Annex 5: Speed and Distance Monitoring, Model in SCADE Suite

**Explicative document related to SCADE Model so-called “Essai”.**

**The model computes the curves “Vebd”, “Vsbd” and “Vwarning” from data furnished by balise or radio messages and pre-computed in the block A2 (“To check and store balise content” ( - cf diagram A0). Then the model computes and command “EB”, “SB” and “Warning” orders related to train status (speed, position, acceleration…).**

**The computation processus is composed of two parts :**

* **M100\_Precalcul correspond to bloc A4 (“to store track description in Database and Set of Variables”). This bloc computes the curves Vebd, Vsbd & Vwarning that are independant from train position and speed, but may depend on train and track characteristics. These curves need to be up-dated at each time these characteristics change ( that means : change of LRBG, TSR, route, MA, etc..).**
* **M200\_Supervision correspond to bloc A5 (“to achieve processes). This bloc command fail-safe orders thanks to curves elaborated from bloc M100\_Precalcul and from train status (position, speed etc..).**

**Assumptions :**

*1. Inputs Data :*

* The receipt of data is only profiles:
* That is “set” { di ,Vi } indicating that a value « Vi » is operational since one distance « di », all distances having a common origin so-called LRBG.
* The values « Vi » can be SSP, TSR, Gradients, EOA, DP etc…
* Inputs data are refreshed by « bloc A4 » at each change of any data, including the change of LRBG.
* Train Length is taken into account by bloc A4, by shifting position of SSP or Gradient when increasing.
* Example is shown hereafter :

Gd

D1

D1+ L\_TRAIN

D2

D

- 0.3 %

+ 0.1 %

- 0.5 %

***2. Technical Choice of « Step » for computation :***

These values do not permit computations as they are received. They need to be put in coherence within a database, based on a description of « step by step ». Three methodologies are possible:

* « steps » basic of « x » meters, constant (recommended),
* « steps » of increasing size following a mathematic law,
* « steps » of random size following changes of data inputs.

The model can use « steps » basic of « x » meters, constant or « steps » of increasing size following a mathematic law

**M000\_Root:**

**M100\_PreCalcul:**

M100 is computed only one time, when some variables have changed:

To each step is attached elementary variables organised in table:

* Header composed of:
* step number: numbering from 1 up to “N”,
* step position, from 0 = LRBG Position up to DP position (at least) = portion de voie allouée au train,
* step size (step position + step size = next step position).
* Input Data:
* SSP value (1), coming from SSP packet,
* Gradient value (1), coming from Gradient packet,
* EOA and DP position (2), coming from MA packet,
* TSR value (3), coming from TSR packet(s),
* Adh indication (1), coming from Adh packet.
* Computed Data:
* MRSP, computed by M120\_MRSP Operator,
* dV value, computed by M122\_dV\_Operator,
* Asafe value, computed by M130\_ACCeler\_Operator,
* Aservice value, computed by M130\_ACCeler\_Operator,
* AEB value, computed by M132\_AEB\_Operator,
* ASB value, computed by M\_133\_ASB\_Operator,
* Vebd value, computed by M140\_EBD\_Operator ,
* Vsbd value, computed by M140\_EBD\_Operator,
* Vwarning value, computed by M140\_EBD\_Operator,
* Mode of supervision (T, C, R), computed by M140\_EBD\_Operator

Table can be organized in step of variable size or fixed size.

**M110\_Step\_Operator:**

The distance from LRBG up to EOA is split into 110 steps,

All steps have a size in geometric progression (reason = 1.06),

Initial size of step 1 is 1.06 m, Max covered distance is 10109,64m.

**M120\_MRSP\_Operator:**

MRSP operator is grouping all data related to speed,

All data of type (d, v) is transformed into variables attached to steps,

“d” being the distance from reference where “v” must be applied,

“v” being the value to be applied (speed, gradient, Adh, EOA).

MRSP operator is using **“Spreader\_Operator”** to deal data related to speed,

dV allowance are dealt through **“dV\_Operator”**.

Explanation in English:

The operators M\_120 and M\_130 are scanning the database in order to elaborate 1 vector by type of data (SSP, TSR, Adh, Gradient, etc..)

step 1

step 2

step 3

step 4

step xx

step xy

Da, Va

Db, Vb

Dc, Vc

**In the example hereafter, the scanning give the final result as follows:**

* Step 1 and Step 2: init value,
* Step 3: Va,
* Step 4: Va,
* Step xx: Vc,
* Step xv: Vc.

As result, the algorithm is:

For each « step » of ranking i = 1 to N:

* Deb:= Position(i)
* End:= Position(i+1)
* For each singularity {Dj, Vj} of ranking j = 1 to n:
* EXIT IF ( D(j) > End) << singularity over the step >>
* IF (D(j) < Deb THEN M(i):= M(i-1) << yet taken in account >>
* ELSE M(i):= V(j) << singularity to be stored >>

**Important precondition: D(j+1) >= D(j)**

**M130\_ACCeler\_Operator:**

ACCeler operator is grouping all data related to A\_safe and A\_service,

ACCeler operator is using:

**Boolean Spreader operator** (to take into account Adh),

**AEB and ASB operators** (to take into account train characteristics),

**Compute\_Acc** (to take into account slope and gravity).

**M140\_EBD\_Operator:**

EBD operator is grouping all data related to Emergency Braking curves,

Including Vsbd and Vwarning when deduced from Vebd by timer,

must be computed in reverse (from EOA to LRBG).

**M150\_SBD\_Operator (TBD)**:

SBD operator is grouping all data related to Service Braking curves (SBI1),

involved when Vsbd is not deduced from Vebd (SBI1 used).

must be computed in reverse (from EOA to LRBG),

**M200\_Supervision:**

**M210\_Delta\_Operator:**

Delta\_Operator estimates speed and position at EB start {ANT\_V, ANT\_P0},

then with same speed, estimates position at SB start (ANT\_P1),

then with same speed, estimates position at W start (ANT\_P2).

**M220\_Search\_Operator:**

Search\_Operator is used to find the step located at a given location,

Provides speed related to two positions:

actual position (TRAIN\_POS)

anticipated position:

ANT\_POS\_0 ( for EB),

ANT\_POS\_1 ( for SB, 5 seconds after EB)

ANT\_POS\_1 ( for W, 5 seconds after SB).

**M230\_Target\_Operator:**

Target\_Operator is used to find the end of ceiling area in reverse direction,

Should be put in precalcul.

## Préconisations

- Use « step » of constant size. The management of variable size is more complex and require the following capabilities :

- Sorting following increasing position,

- Interpolation within step,

- Sequential access instead of direct access.

- For reducing duration of computation :

* To reduce quantity of steps : for instance : steps of 1m up to 1000 m ahead train position and then 10 m overhead. For covering 10.000 meters of MA, only 1900 steps are required instead of 10.000 steps.
* To replace Control Speed (that is Vebd) by Control Energy (that is Vebd²) in order to avoid frequent “square root” computation.

# Annex 6: « steps » of random size

A step of random size require to use specific calculations.

Therefore, two methods of ***“sorting”*** and one method of **“*interpolation*”** are exposed hereafter :

## 1. Method of sorting

T: table of data structure composed of:

* T.Position: Distance from origin
* T.Type: Type of singularity (speed, gradient, adhesion,..)
* T.Value: Associated value (speed, gradient, adhesion,..)
* T.Size: Distance to next singularity (to be calculated after sorting)

Nb: size of T or number of steps,

Ind: index of singularity,

i, j, k: index of scanning.

**Algorithm Method 1: “Successive Minimum”**

To start from T[1] with i = 1,

To search index “Ind” of T.Position minimum within table,

To exchange T[i] with T[Ind],

To repeat from T[2] up to T[Nb], (Nb-1 should be enough),

Algorithm is as follows:

FOR i:= 1 TO Nb DO <<scan table “T” from 1 to Nb>>

Ind:= i <<initialize mini with 1st value>>

FOR j:= i TO Nb DO <<scan sub-table from i to Nb>>

IF T.Position[j] < T.Position[Ind] THEN Ind:= j END

END <<index Ind is giving mini of sub-table >>

T[i]:=: T[Ind] <<exchange mini of sub-table with “T[i]” >>

END

**Algorithm Method 2: “Bubles Sorting”**

To start from T[0] with i = 0,

To search index “k” from end of T to start of T,

To exchange T[k] with T[k-1] if not ordered,

Algorithm is as follows:

FOR i:= 0 TO Nb-1 DO <<scan table “T” from 0 to Nb-1>>

FOR k:= Nb-1 TO i+1 DO <<scan in reverse “T” from Nb-1 to i+1>>

IF T.Position[k] < T.Position[k-1] THEN T[k]:=: T[k-1] END

END << all T[k] whose position is smaller than previous one is inverted >>

END << algorithm can be shortened if no inversion is operated during a loop “i”>>

## 2. Interpolation

Interpolation is needed in two cases:

* when searching the exact “Vebd” value at a given position “x”:
* A scanning permits to define the step “n” where is located the position “x”,
* computation of “d” permits to conclude:
* (d < 0) or (d = 0): result is MRSP + dV,
* (d > 0) and (d > x): result is MRSP + dV,
* else result is Vx solution 2nd degree equation.
* when searching the exact “Position” value where MRSP curve is meeting the braking curve:
* computation of “d” permits to conclude.
* if (Vebd(n+1) >= Vebd(n)) then d = Size(n).

MRSP + dV

**Vebd(n)**

**Vebd(n+1)**

d

Size(n)

interpolation: d = Size(n) + [Vebd(n+1)² -Vebd(n)²] / (2 \* Asafe(n))

*x*

V(x) solution of: x = Size(n) + [Vebd(n+1)² -V(x)²] / (2 \* Asafe(n))

V(x)

V(x) solution of: V(x)² = { [ ( Size(n) – x) \* 2 \* Asafe(n) ] + [Vebd(n+1)²] }