

Transition	Modeled tasks and gateways	Comments
S0/S1	Tasks: som_start (Engage the procedure)	The driver must engage the procedure using the DMI to start the mission
S1/S2	Tasks: som_enterid_dmi_1 (Enter the Driver-ID) som_retry_dmi_1 (Retry) som_storeid_evc_1 (Store the Driver-ID) som_validate_evc_1 (Validate level and position) Gateways: XOR between som_enterid_dmi_1 som_retry_dmi_1 som_storeid_evc_1	The driver has to enter the Driver-ID, but during the process he/she could fail. The system provides a retry mechanism which is abstracted in BPMN terms with a XOR-split between <i>Enter the Driver-ID</i> , <i>Retry</i> and <i>Store the Driver-ID</i> , which allows for only one activity to be run between <i>Retry</i> and <i>Store the Driver-ID</i>
S1/S4	Tasks: som_enterid_dmi_1 (Enter the Driver-ID) som_retry_dmi_1 (Retry) som_storeid_evc_1 (Store the Driver-ID) som_validate_evc_1 (Validate level and position) Gateways: XOR between som_enterid_dmi_1 som_retry_dmi_1 som_storeid_evc_1	This is basically the same trace as followed by the transition S1/S2, but the On-Board system recognizes the validity of the level and position, preparing the system to contact the RBC.
S2/S3	Tasks: som_validate_evc_1 (Validate level and position)	This same activity takes in account the insertion of the level by the user

S3/S10	<p>Tasks:</p> <p>som_openconn_rtm_1 (Open the connection)</p> <p>som_giveup_evc_1 (Give up)</p> <p>som_checkpos_rbc_1 (Check the position)</p> <p>textbfsom_checktrain_rbc_1 (Check the train)</p> <p>som_storepos_rbc_1 (Store the valid position)</p> <p>som_storevalacc_rbc_1 (Store the 'valid' and 'accepted' flag (RBC))</p> <p>som_storeacc_evc_1 (Store the 'accepted' flag (EVC))</p> <p>som_driversel_dmi_1 (Await driver selection)</p> <p>Gateways:</p> <p>XOR between som_validate_evc_1 som_openconn_rtm_1 som_retry_dmi_2</p> <p>XOR between som_openconn_rtm_1 som_retry_dmi_2 som_checkpos_rbc_1 som_giveup_evc_1</p> <p>XOR between som_checkpos_rbc_1 som_checktrain_rbc_1 som_storepos_rbc_1</p> <p>XOR between som_checktrain_rbc_1 som_storepos_rbc_1 som_storevalacc_rbc_1</p> <p>XOR between som_storeacc_evc_1 som_giveup_evc_1 som_driversel_dmi_1</p>	<p>From S3 the process allows the system to contact the RBC and go through all the checks that need to be done in order to later allow the train to transition to the FS mode. As seen in the state diagram, it is also possible that the level is 0, 1 or NTC, which triggers the <i>Give up</i> activity that models the transition to the driver mode selection. Also, a <i>Retry</i> activity is modeled if the use is interested in retrying to connect to the RBC if the connection initially fails. The activities <i>Store the 'valid' and 'accepted' flag</i> and <i>Store the 'accepted' flag (RBC)</i> are modeled to consider the instances where the train or the position encounter some problems while being assessed.</p>
--------	---	---

S4/S10	<p>Tasks:</p> <p>som_openconn_rtm_1 (Open the connection)</p> <p>som_checkpos_rbc_1 (Check the position)</p> <p>textbfsom_checktrain_rbc_1 (Check the train)</p> <p>som_storepos_rbc_1 (Store the valid position)</p> <p>som_storevalacc_rbc_1 (Store the 'valid' and 'accepted' flag (RBC))</p> <p>som_storeacc_evc_1 (Store the 'accepted' flag (EVC))</p> <p>som_driversel_dmi_1 (Await driver selection)</p> <p>Gateways:</p> <p>XOR between som_validate_evc_1 som_openconn_rtm_1 som_retry_dmi_2</p> <p>XOR between som_openconn_rtm_1 som_retry_dmi_2 som_checkpos_rbc_1 som_giveup_evc_1</p> <p>XOR between som_checkpos_rbc_1 som_checktrain_rbc_1 som_storepos_rbc_1</p> <p>XOR between som_checktrain_rbc_1 som_storepos_rbc_1 som_storevalacc_rbc_1</p> <p>XOR between som_storeacc_evc_1 som_giveup_evc_1 som_driversel_dmi_1</p>	<p>The logic is basically the same as the one explained for the transition S3/S10, with the difference that the <i>Give up</i> activity is not taken in account since, to get to S4, the level must be 2 or 3.</p>
--------	--	--

S10/S12	Tasks: som_inserttraindata_dmi_1 (Insert train data and number)	The transitions S10/S12 and S12/S13 are collapsed in a single activity.
S12/S13	Tasks: som_inserttraindata_dmi_1 (Insert train data and number)	The transitions S10/S12 and S12/S13 are collapsed in a single activity.
S13/S10	Tasks: som_checkrbcsess_rtm_1 (Check RBC session)	If the session is dropped an anomaly is notified to the user. The process then continues from the selection by the driver of the mode, as seen in the state diagram.
S13/S11	Tasks: som_checkrbcsess_rtm_1 (Check RBC session)	If the session is still up the process continues.
S11/S20	Tasks: som_selstart_dmi_1 (Select 'Start')	If the user selects 'Start', he/she has also acknowledged the data transmitted by the RBC.
S20/S24	Tasks: som_checklev_evc_1 (Check level) Gateways: XOR between som_selstart_dmi_1 som_sendMAreq_rtm_1 som_checklev_evc_1	The XOR gateway models the fact that if the level is NTC, 0 or 1, a check on these values must be made to yield the correct mode proposal.
S20/S21	Tasks: som_sendMAreq_rtm_1 (Send MA Request) Gateways: XOR between som_selstart_dmi_1 som_sendMAreq_rtm_1 som_checklev_evc_1	

S21/S24	<p>Tasks:</p> <p>som_checktrainroute_rbc_1 (Check the route for the train)</p> <p>som_checkval_rbc_1 (Check the 'valid' flag)</p> <p>som_grantSR_rbc_1 (Grant SR mode)</p> <p>som_awaitack_dmi_2 (Await driver acknowledgement)</p> <p>Gateways:</p> <p>XOR between</p> <p>som_checkval_rbc_1 som_grantSR_rbc_1 som_grantOS_rbc_1 som_grantFS_rbc_1</p>	<p>The transition is subjected to the check against the already cited parameters that look for the route of the train and the validity of the position of the train traced by the RBC. If some anomalies in the position and the train route are detected, the RBC proposes the SR mode to the driver.</p>
S21/S25	<p>Tasks:</p> <p>som_checktrainroute_rbc_1 (Check the route for the train)</p> <p>som_checkval_rbc_1 (Check the 'valid' flag)</p> <p>som_grantOS_rbc_1 (Grant OS mode)</p> <p>som_awaitack_dmi_1 (Await driver acknowledgement)</p> <p>Gateways:</p> <p>XOR between</p> <p>som_checkval_rbc_1 som_grantSR_rbc_1 som_grantOS_rbc_1 som_grantFS_rbc_1</p>	<p>The logic is the same as the one seen in S21/S24, with the difference that the RBC proposes the OS mode in this case.</p>

S20/S22	Tasks: som_checklev_evc_1 (Check level) som_grantSN_evc_1 (Grant SN) Gateways: XOR between som_checklev_evc_1 som_grantSR_evc_1 som_grantUN_evc_1 som_grantSN_evc_1	This transition is performed when the stored level is NTC.
S20/S23	Tasks: som_checklev_evc_1 (Check level) som_grantUN_evc_1 (Grant UN) Gateways: XOR between som_checklev_evc_1 som_grantSR_evc_1 som_grantUN_evc_1 som_grantSN_evc_1	This transition is performed when the stored level is 0.

Table 1: Mapping between the state transitions and the modeled activities and gateways