

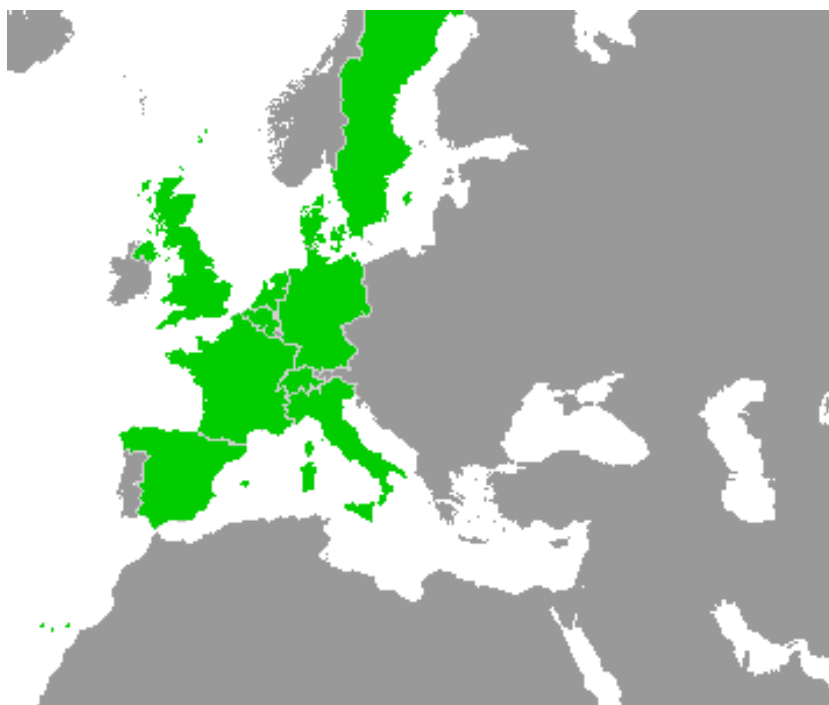
Work-Package 3: "Modeling"

Guidelines for SysML modeling with Papyrus Model Documentation

This document aims at describing the Papyrus tool usage in the frame of the OBU EVC software modelisation.

Alexandre Ginisty

July 2013
Revised August 2013



This page is intentionally left blank

Guidelines for SysML modeling with Papyrus Model Documentation

This document aims at describing the Papyrus tool usage in the frame of the OBU EVC software modelisation.

Alexandre Ginisty

All4tec
Immeuble Odyssee Bâtiment E
2-12, Rue du Chemin des femmes
91 300 MASSY, France

Description of work

This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License.



Abstract: This document provides the methodology and the use of the modeling tool Papyrus, in order to model the OBU EVC Software with UML/ SysML language. It is willing to support the good practices, knowledges, and overall modeling methodology, and spread them out to the project. In accordance with the open source philosophy, this methodology is willing to be transparently shared with all modelers involved in the project, and to be an alive support for modeling. Every modeler is invited to provide additional informations regarding the methods or use of Papyrus, in order improve this document and to make out of it a real user manual for OBU EVC modeling with Papyrus.

Disclaimer: This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 – (cc by-sa 3.0)

THE WORK IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

<http://creativecommons.org/licenses/by-sa/3.0/>

Table of Contents

1	Short Introduction to Formalism and Tool.....	4
2	Prerequisites	4
2.1	Papyrus platform	4
2.2	Importing an existing Papyrus project :	4
3	Modeling Strategy	5
4	Model Overview	6
5	Model Benefits and shortcomings	9

Figures and Tables **Figures**

Figure 1.	Example of diagram used	4
Figure 2.	Menu "Import"	5
Figure 3.	Importing existing project.....	5
Figure 4.	Simplified system structure	6
Figure 5.	OBV behavior for mode switching.....	6
Figure 6.	Sub-statemachine of FS mode.....	7
Figure 7.	Statechart describing procedure from FS point of view	7
Figure 8.	Activity diagram for "Check rectangle of conditions	8
Figure 9.	Modeling choice using SysML	8
Figure 10.	Statechart describing procedure from OS point of view.....	9
Figure 11.	Sub-statemachine describing OS to FS procedure	9

Tables

1 Short Introduction to Formalism and Tool

This document describes the modeling of the On-Sight Procedure. The model has been made from the specification description of the SRS SUBSET-026-3.5 baseline 3 as recommended by D2.5 Methods and tools benchmarking methodology. The formalism used is SysML. More particularly we had used block diagram, state charts and activity diagrams. SysML is a graphical language that extends UML for a customize version suitable for system engineering. It may help modeling system within a board range of system variety that may include hardware, software, data, personnel and facilities. It supports the specification, analysis design, verification and validation of complex system. Papyrus is an open source tool that has been used to implement the model. Papyrus is aiming at providing an integrated and user-consumable environment for editing any kind of EMF model and particularly supporting UML and related modeling languages such as SysML and MARTE. The main feature of Papyrus regarding this latter point is a set of very powerful customization mechanisms which can be leveraged to create user-defined Papyrus perspectives and give it the same look and feel as a native DSL editor. This model is currently not executable or testable but its transformation is scheduled.

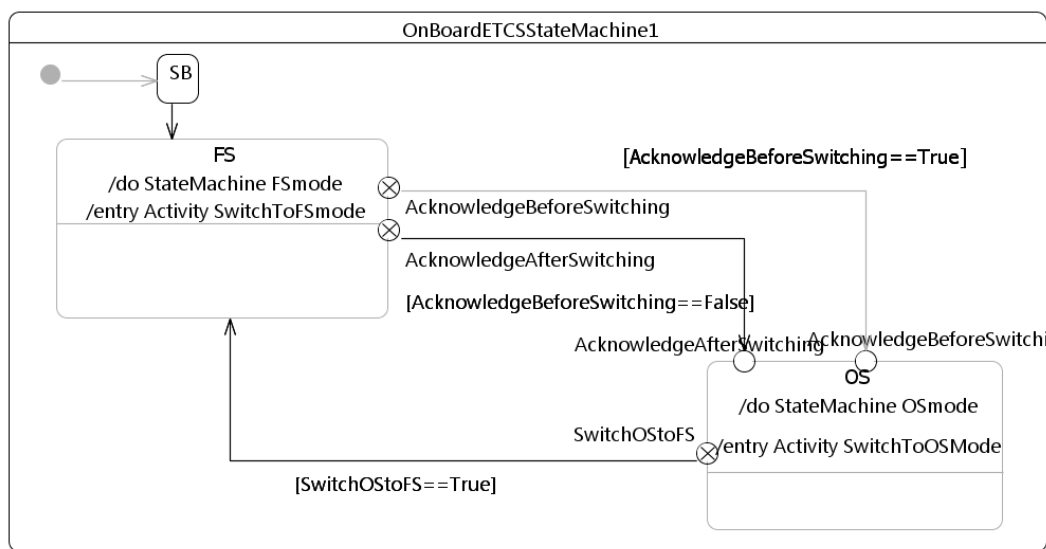


Figure 1. Example of diagram used

2 Prerequisites

This section shortly describes the basic prerequisites to use Papyrus projects on github.

2.1 Papyrus platform

This model has been made on Papyrus under Eclipse platform version Juno available here: <http://www.eclipse.org/downloads/packages/eclipse-modeling-tools/junosr1> To download Papyrus, launch Eclipse, go to “Help -> Install Modeling Component” and select Papyrus.

2.2 Importing an existing Papyrus project :

Papyrus projects are composed of 4 files, “.project”, “.uml”, “.notation” and “.di”. To import an existing Papyrus project, launch Papyrus, click on “File » Import » Existing project into Workspace”.

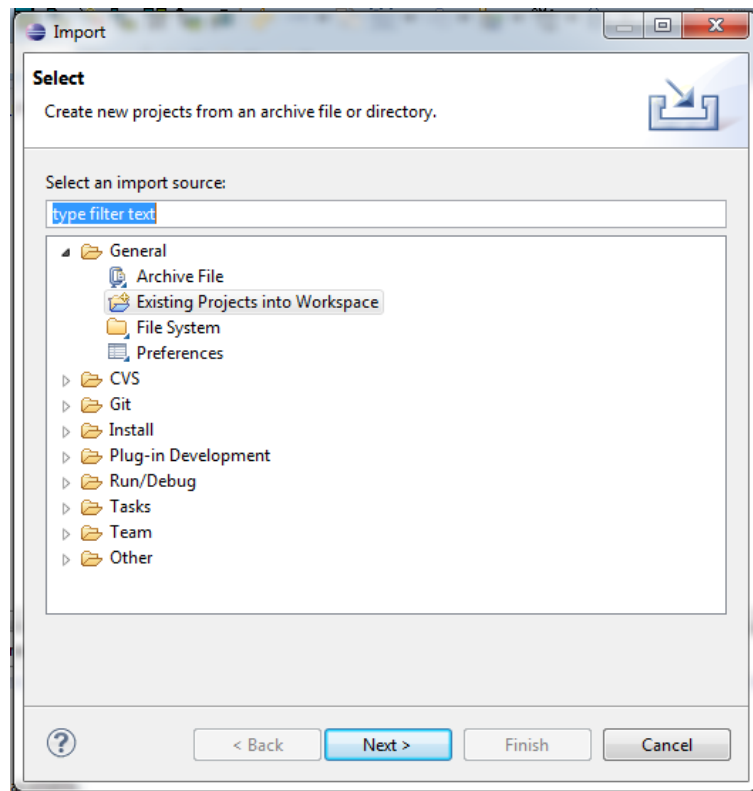


Figure 2. Menu "Import"

Then Browse to the directory to select the project and click on Finish.

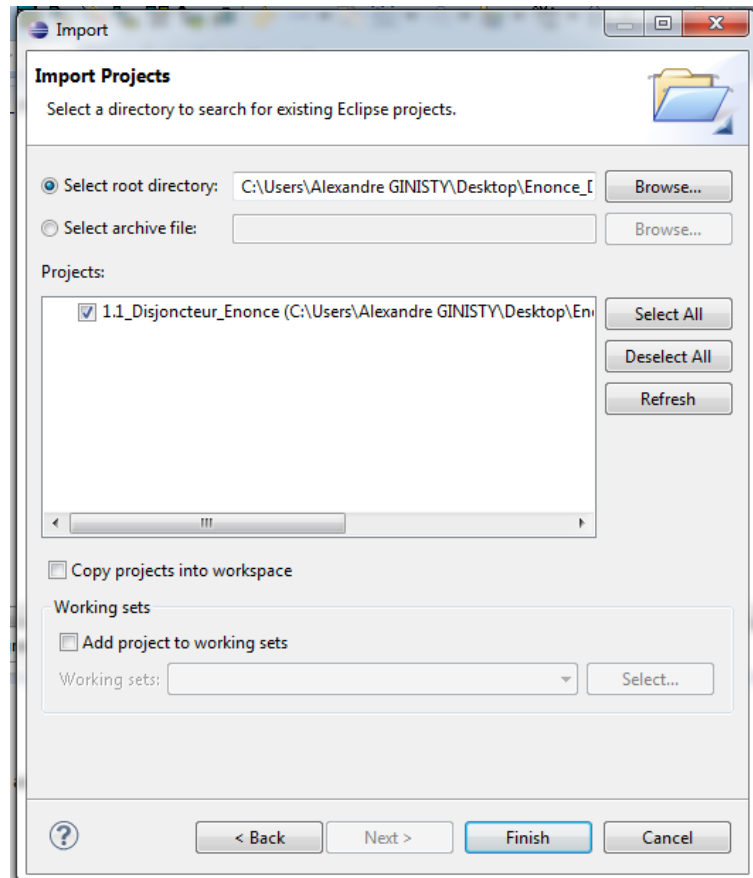


Figure 3. Importing existing project

3 Modeling Strategy

A procedure like the On Sight procedure can be described in first approach in a SysML Sequence Diagram. It gives a better understanding, and allows to pinpoint which parts and which functions are involved in this procedure. Then, based on Subset 026, we made block diagrams to model the whole system. Then we decided that, for a benchmark purpose, a simplified system will be better, and we modeled it. The current model uses 3 main parts, the on-board unit, the rbc and the driver. We decided to focus on the on-board unit to model its behavior. We also decided to choose the initial mode and the ending mode of the procedure, and we don't consider any level change during the procedure.

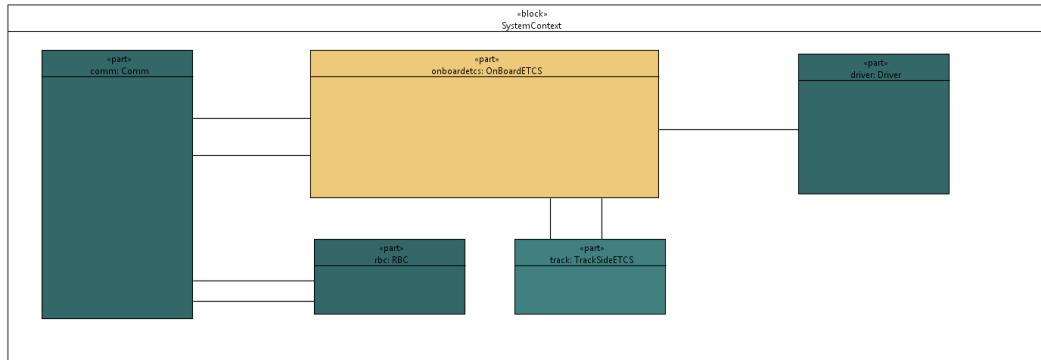


Figure 4. Simplified system structure

4 Model Overview

There are 4 packages in the model but only two are used at the moment, the system model package and the System Requirements package. The procedure is mainly described with statecharts. We started with sequence diagrams and activity diagrams, but found that the use of statecharts fill much our needs and can be more useful for further development (i.e using plugins for V&V). The procedure involves trackside means (RBC, Euroloop, etc.), the On Board unit (OBU) and the driver. On Sight mode request may come from trackside means, considering we can only be in Full Supervision mode when it comes (model restriction). In our model we only consider messages exchanged from OBU point of view.

The overall behavior of On Board unit is modeled as below:

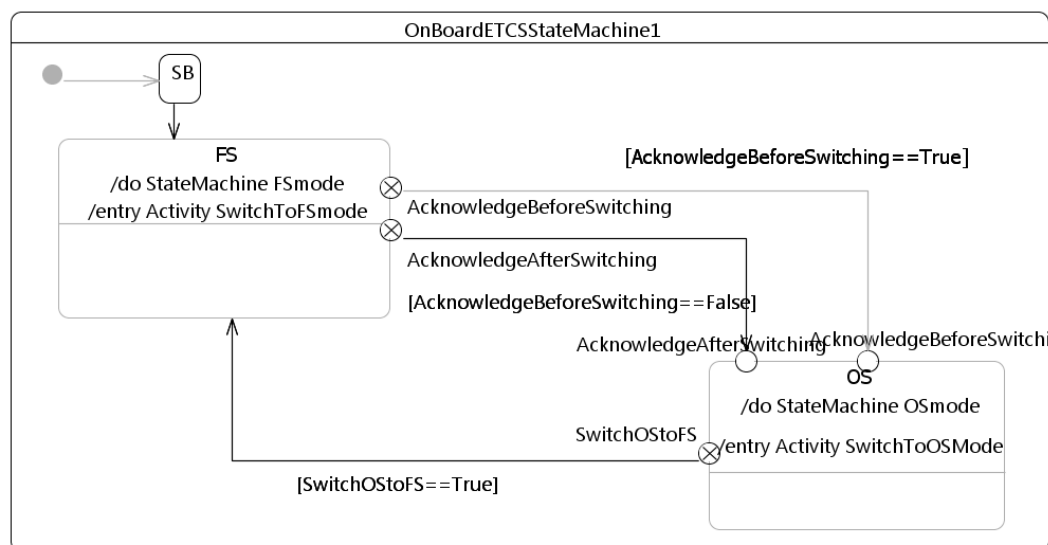


Figure 5. OBU behavior for mode switching

Starting from SB mode for consistency, the OBU switch to FS mode.

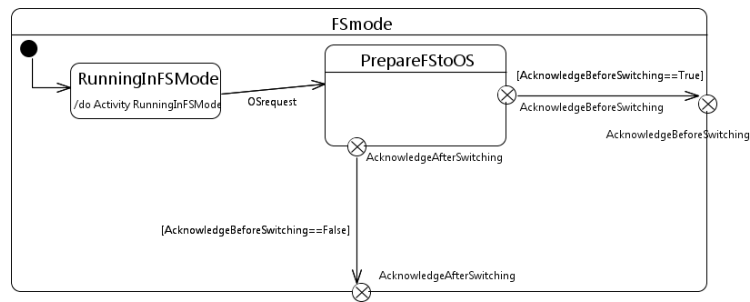


Figure 6. Sub-statemachine of FS mode

This Statechart models the first part of the procedure which concerne the OBU while it is in FS mode (OS request, Check conditions of displaying OS acknowledgement request, etc.)

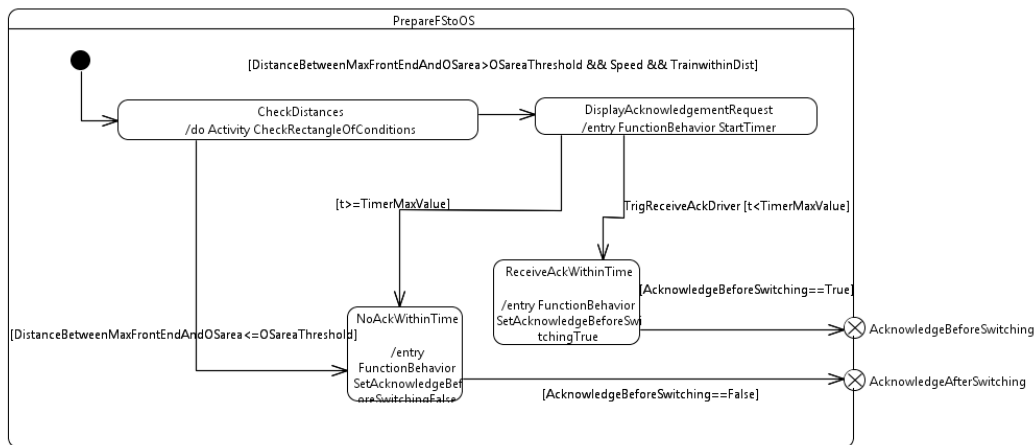


Figure 7. Statechart describing procedure from FS point of view

Activity diagrams are used to model low-level operations, such as checking the “rectangle of conditions”. The use of “Opaque Operation” has been made in order to model these operations without knowing exactly what they actually do.

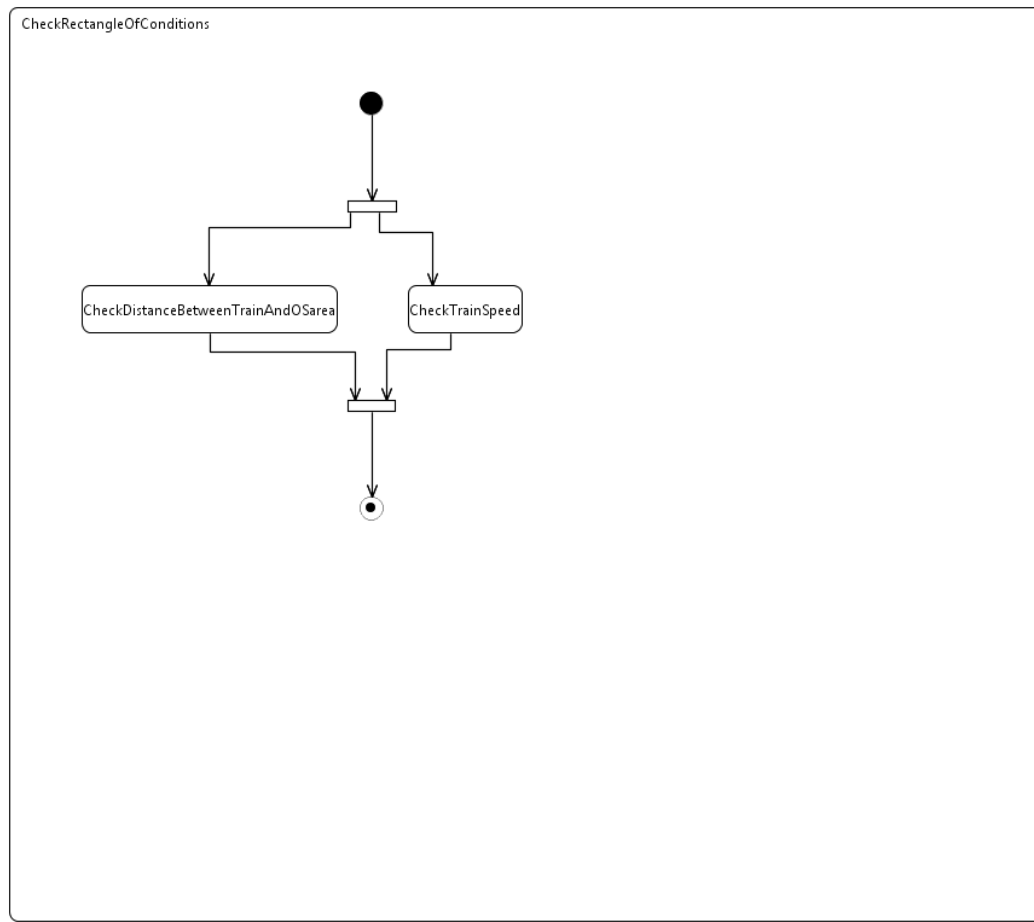


Figure 8. Activity diagram for "Check rectangle of conditions"

Notice that we decided to use the method of modeling that consider a distribution of the procedure between concerned modes, as it is described in Munich presentation (https://github.com/openETCS/model-evaluation/blob/master/Munich_20130416/c_Papyrus_SysML_V1.pdf)

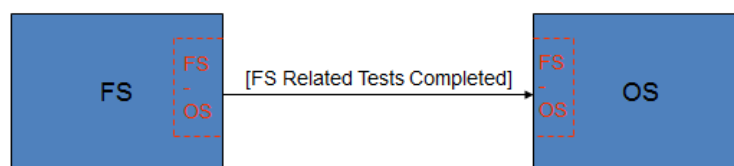


Figure 9. Modeling choice using SysML

Operations can be done during the entry or the exit of a state, or while the concerned state is active. Multiple choices are available for the location of the different operations and it needs experimentation to find which ones fit the most to reality. Two conditional exits are possible, depending of the driver acknowledgement. They lead to two conditional entries in OS mode (OS Statechart).

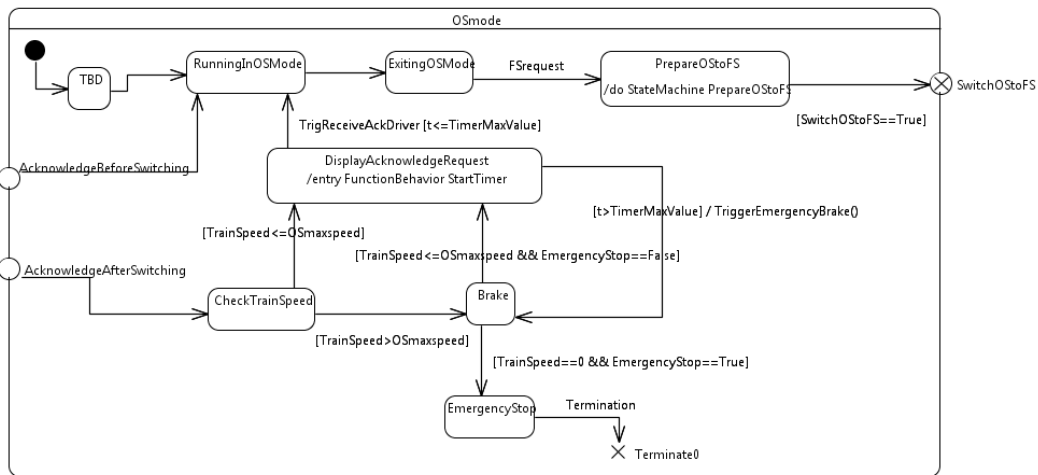


Figure 10. Statechart describing procedure from OS point of view

In the same way, the procedure to switch from OS mode to FS mode is located in both modes (in fact, only in OS mode because of the procedure itself as it is described in Subset 026).

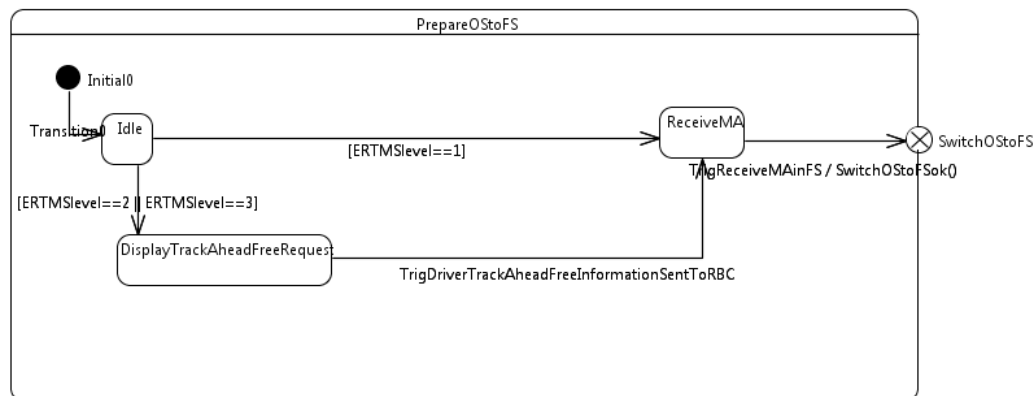


Figure 11. Sub-statemachine describing OS to FS procedure

Diagrams and states can be easily linked to other parts of the subset 026 related to « procedures », such as receiving a new MA (which can also be described using statecharts and activity diagrams). This insure that further development and integration is possible (considering using only SysML for that level of models).

5 Model Benefits and shortcomings

We think that using SysML provides these benefits:

- Graphical language, easy to understand with few explanations, useful for demonstration,
- Requirement and structure traceability,
- Facilitate proving behavior consistency at high level.

Also, using SysML with Papyrus provides these benefits:

- Open Source tool,
- Eclipse based, allowing many plugins to interface with,

- Constant improvement.

However, both SysML and Papyrus have their shortcomings. SysML is a semi-formal language that is aimed at modeled systems at a high level. It allows many means of modeling that can have strong consequences for performances or even system behavior. That choices have to be defined at the beginning of the model to insure consistency between all the parts and all the modelers in SysML. Papyrus, as it is based on Eclipse platform, can have performance problems when modeling big systems. Moreover, the GUI is not very user friendly at the moment and it can be difficult to model efficiently without a good basic knowledge of the tool.