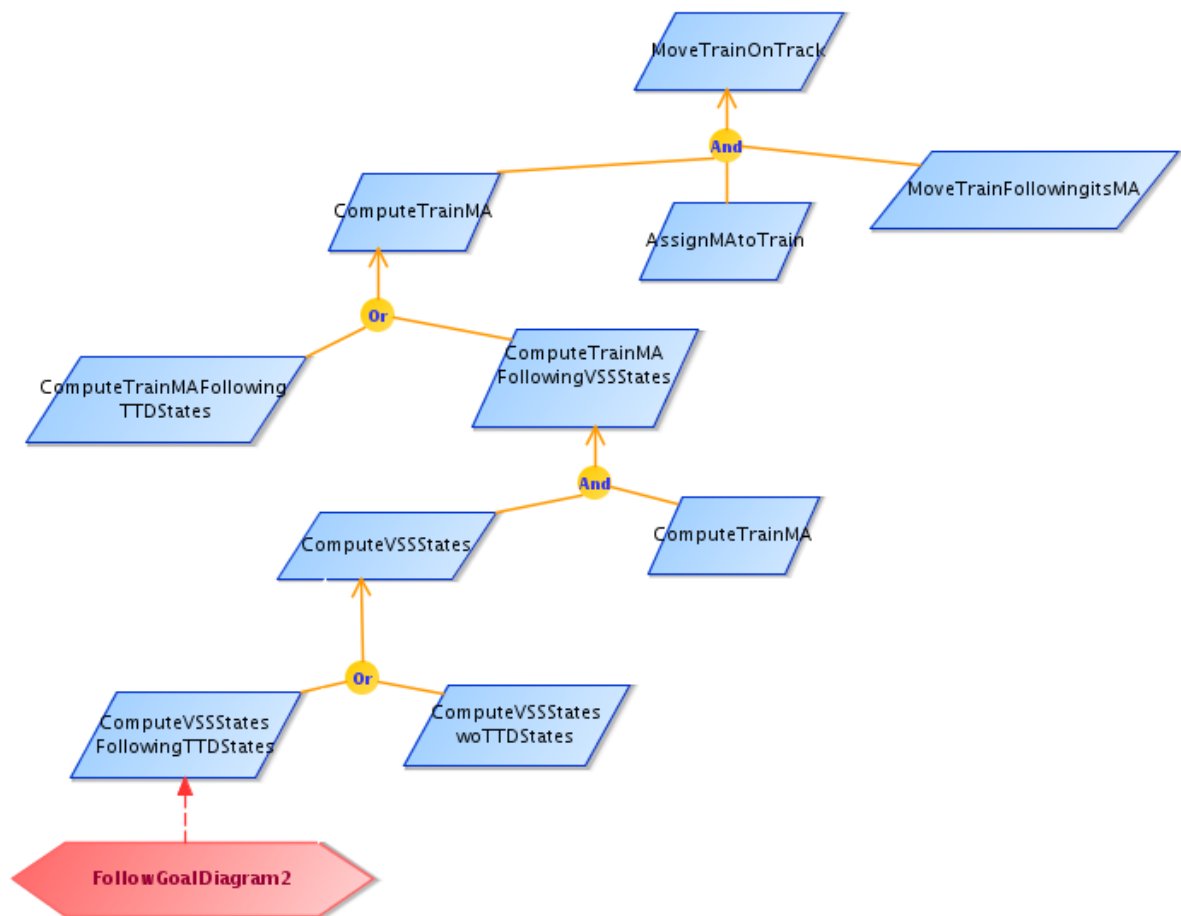


Hybrid ERTMS/ETCS Level 3 Case Study

Goal Diagrams Description

- La voie est matérialisée par un sous ensemble contigue de N : $a..b$ avec $\{a, b\} \subset N$ et $a < b$
- La voie est divisée en portion contiguës appelées TTDs et chaque TTDs en portions contiguës appelés VSSs. $stateTTD \in TTDs \rightarrow \{OCCUPIED, FREE\}$ représente l'état des ttds. Cette fonction fournit l'état courant du TTD. De plus, si un train est présent sur le TTD, alors l'état du ttd retourné par $stateTTD$ sera *OCCUPIED*. De même $stateVSS \in VSSs \rightarrow \{OCCUPIED, FREE, UNKNOW, AMBIGUOUS\}$ représente l'état des vss. A l'état initial, chaque vss est à l'état *unknow*.
- $ttdOfVSS \in VSSs \rightarrow TTDs$ permet de retrouver le ttd qui contient un vss.
- Nous définissons *TRAIN* comme étant l'ensemble des trains et $connectedTrain \in TRAIN \rightarrow BOOL$ matérialisant, pour chaque train que le système a détecté sur la voie, s'il est toujours connecté ou pas. $TRAIN \setminus dom(connectionTrain)$ désigne l'ensemble des trains que le système n'a jamais détecté. En l'occurrence, il s'agit des trains qui ne se sont jamais connectés au système.
- Le train est matérialisé sur la voie par la position estimée de son début (*front*) et la position estimée de sa fin (*rear*), sachant que $front \in dom(connectionTrain) \rightarrow a..b$, $rear \in dom(connectionTrain) \rightarrow a..b$ et $\forall tr \in dom(rear). front(tr) > rear(tr)$:
 - Si *tr* est équipé du système ERTMS et pas du système TIMS, alors $tr \in dom(front) \wedge tr \notin dom(rear)$. On peut supposer dans ce cas, que le train s'achève à la fin du TTD où est localisé son front (section 2 du document de spécifications).
 - Si *tr* est équipé des systèmes ERTMS et TIMS (integer), alors $tr \in dom(front) \cap dom(rear)$
- La *Movement Authorities (MA)* du train représente une portion de la voie qu'il peut considérer comme sienne et sur laquelle il peut circuler librement. $MA \in dom(connectionTrain) \rightarrow \mathcal{P}(a..b) \wedge \forall tr \in dom(connectionTrain). MA(tr) = p..q \subset a..b$. On peut supposer, d'après la description du cas d'étude que tout train respectera toujours sa MA : $\forall tr \in dom(MA). front(tr) \in MA(tr) \wedge (tr \in dom(rear) \Rightarrow rear(tr) \in MA(tr))$

1- Diagramme de buts global:

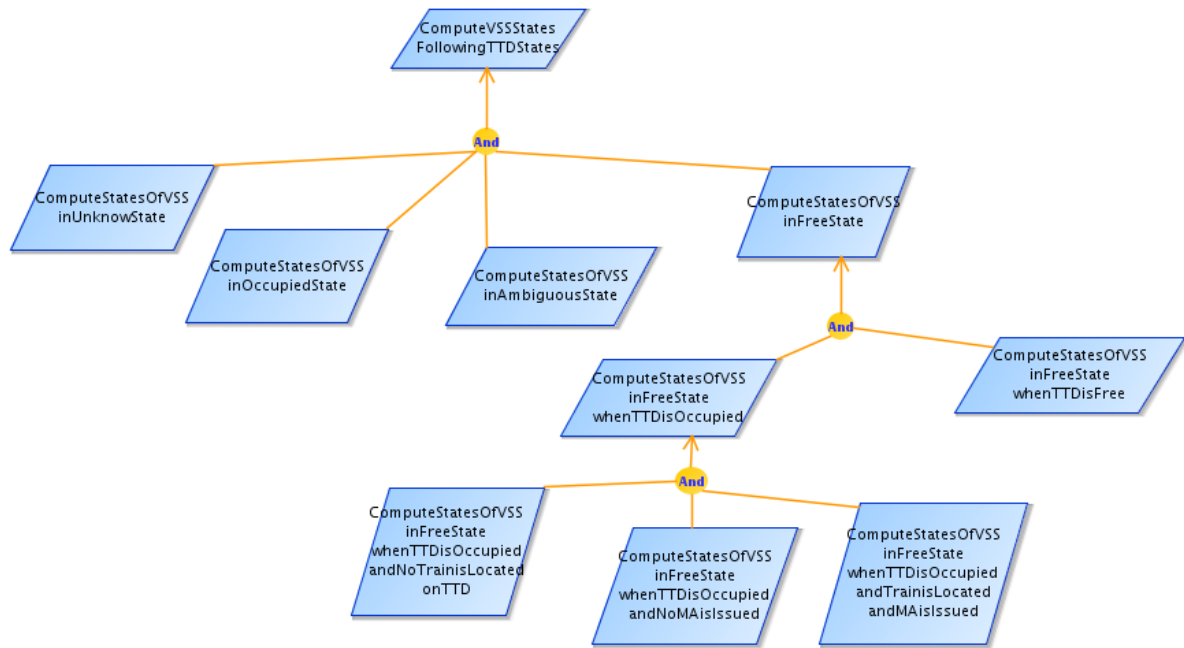


- **MoveTrainOnTrack** : Move train on track:
 - **Paramètres**: $tr \in connectedTrain^{-1}[\{TRUE\}]$, $len \in N1$
 - $front(tr) := front(tr) + len$
 - IF $tr \in dom(rear)$ THEN $rear(tr) := rear(tr) + len$
- **ComputeTrainMA** : Compute the new MA of the Train
- **AssignMAtoTrain**: Assign a MA to a train:
 - **Paramètres**: $tr \in connectedTrain^{-1}[\{TRUE\}]$, $min..max \subset a..b$ ($min < max$)
 - $MA(tr) := min..max$
- **MoveTrainFollowingitsMA** : Move the train on track following its MA :
 - **Paramètres**: $tr \in connectedTrain^{-1}[\{TRUE\}]$, $len \in N1$
 - IF $front(tr) + len \in MA(tr)$ THEN
 - $front(tr) := front(tr) + len$
 - IF $tr \in dom(rear)$ THEN $rear(tr) := rear(tr) + len$
- **ComputeTrainMAFollowingTTDStates** : Compute the MA of a train based on ttd states
- **ComputeTrainMAFollowingVSSStates**: Compute the MA of a train based on vss states
- **ComputeVSSStates**: Compute the current states of vss
- **ComputeTrainMA** : Compute the MA of trains based on current states of vss

- **ComputeVSSStatesFollowingTTDStates**: Compute the current states of vss considering the current states of ttds
- **ComputeVSSStateswoTTDStates** : Compute the current states of vss without considering the current states of ttds

2- Diagramme de buts spécifique au calcul de l'état des VSS :

La référence ici est le diagramme d'états-transitions des VSS (page 24). Nous détaillerons juste la transition #1A afin de valider l'approche.



- **ComputeStatesOfVSSinUnknowState**: compute the current states of vss that were previously in the unknow state
- **ComputeStatesOfVSSinOccupiedState**: compute the current states of vss that were previously in the occupied state
- **ComputeStatesOfVSSinAmbiguousState**: compute the current states of vss that were previously in the ambiguous state
- **ComputeStatesOfVSSinFreeState**: compute the current states of vss that were previously in the free state:
 - o **Paramètres**: $vss = stateVSS^{-1}[\{FREE\}]$, $vss1, vss2, vss3, vss4$
 - o IF $partition(vss, vss1, vss2, vss3, vss4)$ THEN
 - $stateVSS := stateVSS < +(vss1 \times \{OCCUPIED\} \cup vss2 \times \{FREE\} \cup vss3 \times \{AMBIGUOUS\} \cup vss4 \times \{UNKNOWN\})$
- **ComputeStatesOfVSSinFreeStateWhenTTDisFree** : compute the current states of vss that were previously in the free state and for which the ttd is free:
 - o **Paramètres**: $vss2 \subset stateVSS^{-1}[\{FREE\}]$
 - o IF $stateTTD[ttdOfVSS[vss2]] = \{FREE\}$ THEN
 - $stateVSS := stateVSS < +(vss2 \times \{FREE\})$
- **ComputeStatesOfVSSinFreeStateWhenTTDisOccupied**: compute the current states of vss that were previously in the free state and for which the ttd is occupied

- **ComputeStatesOfVSSinFreeStateWhenTTDisOccupiedandNoTrainisLocatedonTTD:**
 compute the current states of vss that were previously in the free state and for which the ttd is occupied with no train located on the ttd:
 - **Paramètres:** $vss4 \subset stateVSS^{-1}[\{FREE\}]$
 - IF $stateTTD[ttdOfVSS[vss4]] = \{OCCUPIED\} \wedge \forall tr \in dom(connectedTrain), ttd \in ttdOfVSS[vss4]. ($
 $ttd = p..q \Rightarrow ($
 $front(tr) < p \vee ($
 $(tr \in dom(rear) \wedge$
 $rear(tr) > q$
 $\vee (front(tr) > q)) // \text{pour tout train dont le rear est}$
 $\text{inconnu, on peut supposer qu'il s'achève à la fin du ttd où son}$
 $\text{front est localisé}$
 $)$
 $)$ THEN
 - $stateVSS := stateVSS < +(vss4 \times \{UNKNOWN\})$
- **ComputeStatesOfVSSinFreeStateWhenTTDisOccupiedandNoMAisIssued:** compute the current states of vss that were previously in the free state and for which the ttd is occupied with no MA issued to a subset of ttd :
 - **Paramètres:** $vss4 \subset stateVSS^{-1}[\{FREE\}]$
 - IF $stateTTD[ttdOfVSS[vss4]] = \{OCCUPIED\} \wedge \forall tr \in dom(connectedTrain) \cap dom(MA), ttd \in ttdOfVSS[vss4]. (MA(tr) \cap ttd = \emptyset)$
 THEN
 - $stateVSS := stateVSS < +(vss4 \times \{UNKNOWN\})$
- **ComputeStatesOfVSSinFreeStateWhenTTDisOccupiedandTrainisLocatedandMAisIssued:** compute the current states of vss that were previously in the free state and for which the ttd is occupied, knowing that there is a train located on the ttd with an MA issued