# **Projet Bonaventure** : Modélisation formelle des exigences d'un système de transport intelligent

Steve Tueno[1,2], Marc Frappier[1], Régine Laleau[2]

[1]GRIL – Université de Sherbrooke, Canada
[2]LACL – Université Paris Est Créteil Val de Marne, France

7 novembre 2018

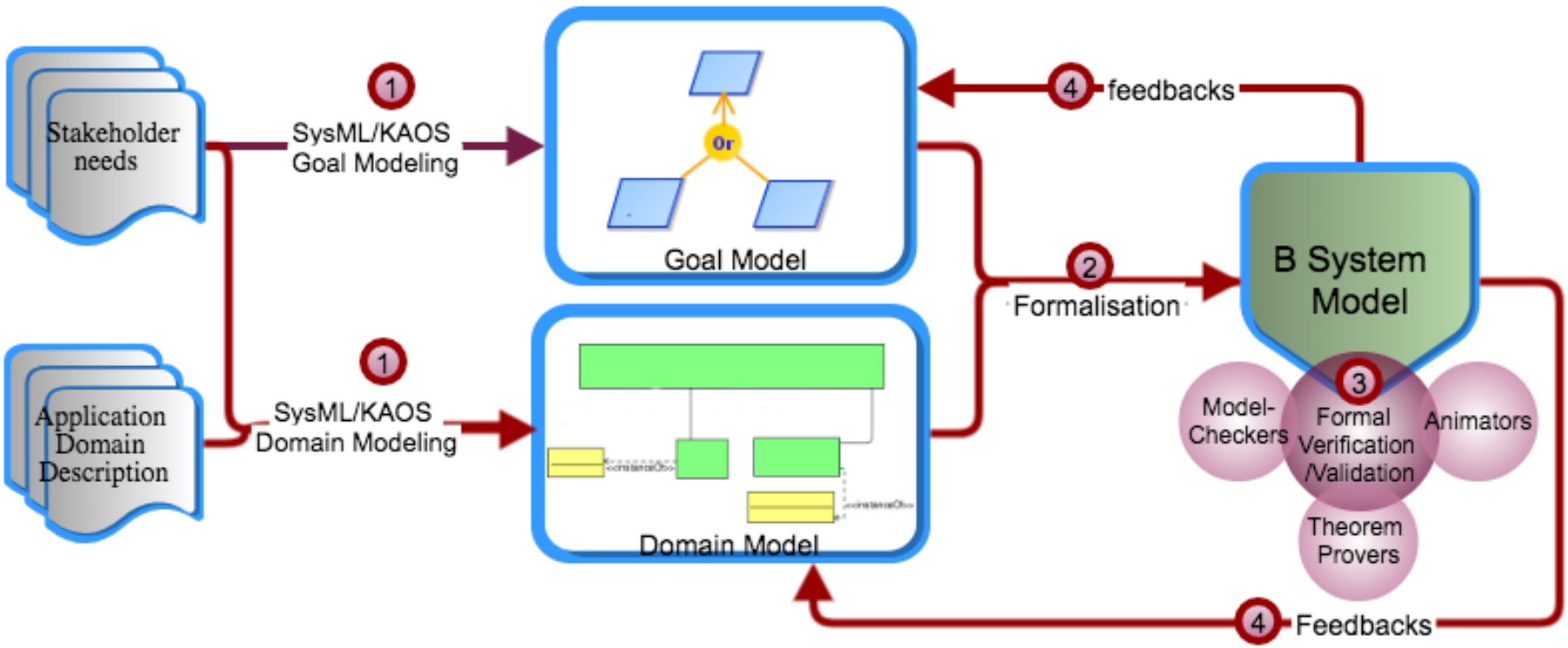
# Sommaire

# SysML/KAOS : Méthode formelle d'ingénierie des exigences

## Projet FORMOSE (ANR-14-CE28-0009)

### FORMOSE :

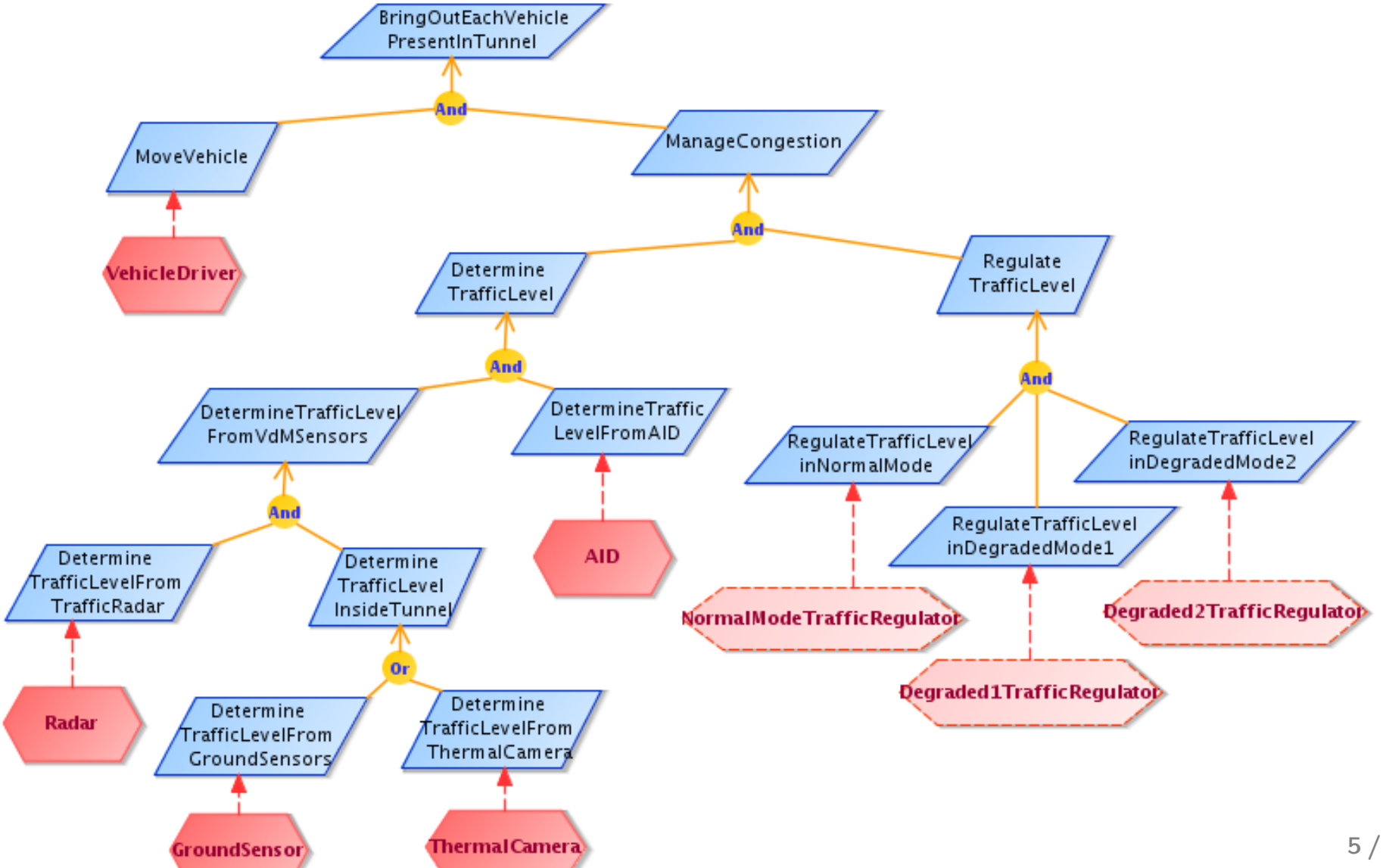
**Méthode** et **outils** pour la **modélisation formelle** des **exigences** de systèmes critiques et complexes.



# Sommaire

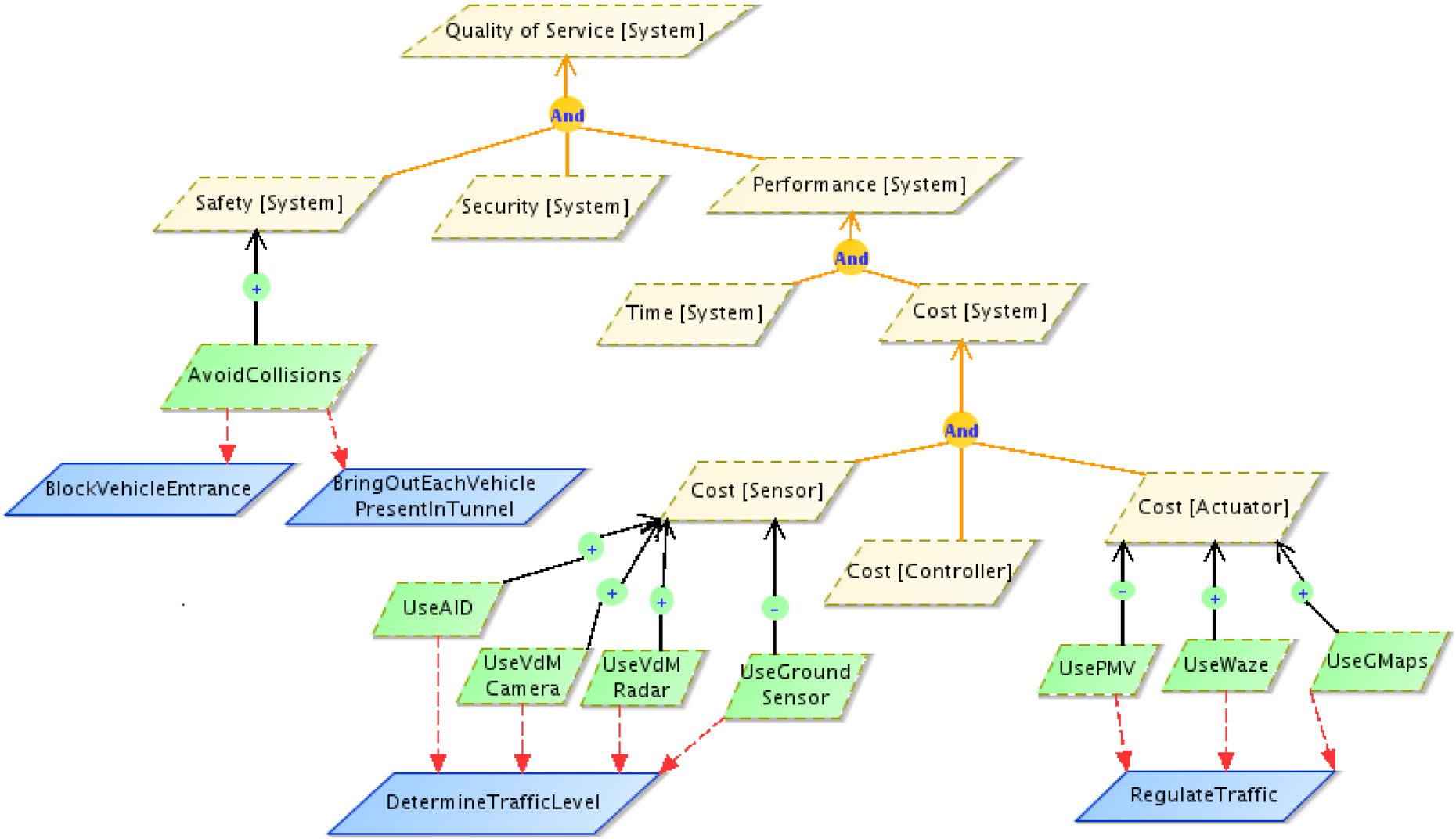
# Modélisation des exigences fonctionnelles

## Livrable : Projet_Bonaventure_Livrable1_functional_goals.pdf



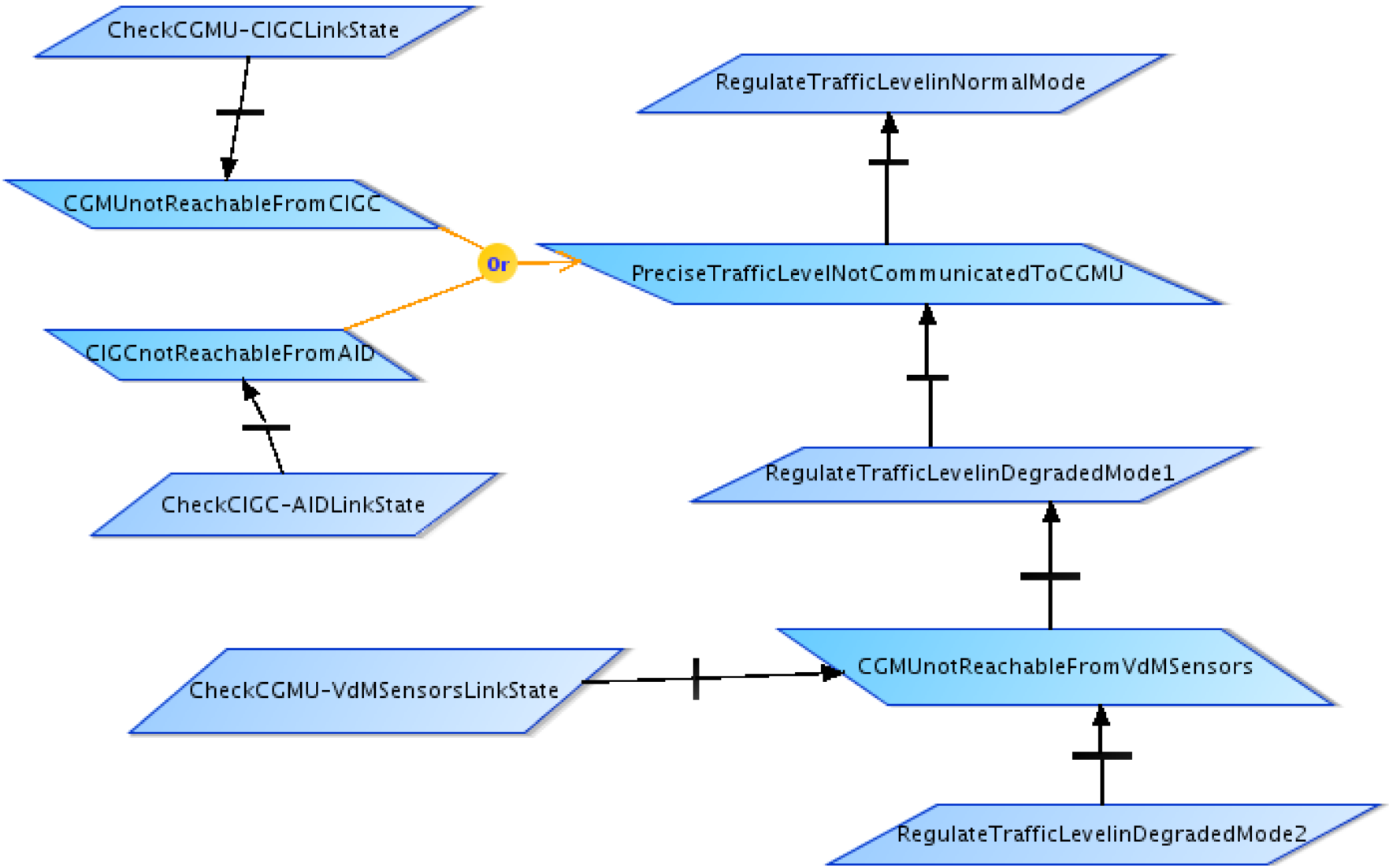
# Modélisation des exigences non-fonctionnelles

## Livrable : Projet_Bonaventure_Livrable2_non_functional_goals.pdf



# Modélisation des obstacles

## Livrable : Projet_Bonaventure_Livrable2_non_functional_goals.pdf



# Sommaire

# Tâches

1. **Définition des objectifs** d'une spécification formelle des exigences du système.
2. **Élaboration d'un modèle de domaine** circonscrit aux objectifs identifiés au point (1).
3. **Construction d'une spécification formelle** circonscrite aux objectifs identifiés au point (1).
4. **Vérification et validation** de la spécification formelle.

# Tâches

1. **Définition des objectifs** d'une spécification formelle des exigences du système.
2. **Élaboration d'un modèle de domaine** circonscrit aux objectifs identifiés au point (1).
3. **Construction d'une spécification formelle** circonscrite aux objectifs identifiés au point (1).
4. **Vérification et validation** de la spécification formelle.

# Tâches

1. **Définition des objectifs** d'une spécification formelle des exigences du système.
2. **Élaboration d'un modèle de domaine** circonscrit aux objectifs identifiés au point (1).
3. **Construction d'une spécification formelle** circonscrite aux objectifs identifiés au point (1).
4. **Vérification et validation** de la spécification formelle.

# Proposition I (1/2)

## Vérification des contraintes de raffinement des buts fonctionnels

### Objectif

**Spécifier formellement les buts fonctionnels** et vérifier :

- **Raffinement AND** : la conjonction des sous-buts satisfait le but parent.
- **Raffinement OR** : la disjonction des sous-buts satisfait le but parent.

Ceci permet de détecter :

- Des **omissions** : les sous-buts ne sont pas suffisant pour satisfaire le but parent.
- Des **ambiguités** : il n'est pas possible de décrire un but de façon précise.
- Des **redondances** : plusieurs buts satisfont le même objectif.
- Des **contradictions**.

# Proposition I (1/2)

## Vérification des contraintes de raffinement des buts fonctionnels

### Objectif

**Spécifier formellement les buts fonctionnels** et vérifier :

- **Raffinement AND** : la conjonction des sous-buts satisfait le but parent.
- **Raffinement OR** : la disjonction des sous-buts satisfait le but parent.

Ceci permet de détecter :

- Des **omissions** : les sous-buts ne sont pas suffisant pour satisfaire le but parent.
- Des **ambiguités** : il n'est pas possible de décrire un but de façon précise.
- Des **redondances** : plusieurs buts satisfont le même objectif.
- Des **contradictions**.

# Proposition I (1/2)

## Vérification des contraintes de raffinement des buts fonctionnels

### Objectif

**Spécifier formellement les buts fonctionnels** et vérifier :

- **Raffinement AND** : la conjonction des sous-buts satisfait le but parent.
- **Raffinement OR** : la disjonction des sous-buts satisfait le but parent.

Ceci permet de détecter :

- Des **omissions** : les sous-buts ne sont pas suffisant pour satisfaire le but parent.
- Des **ambiguités** : il n'est pas possible de décrire un but de façon précise.
- Des **redondances** : plusieurs buts satisfont le même objectif.
- Des **contradictions**.

# Proposition I (2/2)

## Vérification des contraintes de raffinement des buts fonctionnels

**Preuves à décharger :** But $G$ **raffiné** en sous buts $G1$ et $G2$ :

### Opérateur *AND*

- $G_1\_Guard \Rightarrow G\_Guard$
- $G_2\_Guard \Rightarrow G\_Guard$
- $(G_1\_Post \wedge G_2\_Post) \Rightarrow G\_Post$

### Opérateur *OR*

- $G_1\_Guard \Rightarrow G\_Guard$
- $G_2\_Guard \Rightarrow G\_Guard$
- $G_1\_Post \Rightarrow G\_Post$
- $G_2\_Post \Rightarrow G\_Post$
- $G_1\_Post \Rightarrow \neg G_2\_Guard$
- $G_2\_Post \Rightarrow \neg G_1\_Guard$

# Proposition II (1/2)

Vérification de la **concordance** entre **limite de visibilité** et **vitesse maximale autorisée**

## Objectif

Spécifier formellement :

- le **tunnel** ;
- la **courbure du tunnel** : limite de visibilité en chaque point ;
- la **vitesse limite** en chaque point ;

et **vérifier** qu'en chaque point du tunnel, **la vitesse limite soit définie** de façon à donner **suffisament de latitude à un véhicule pour freiner**, dès le moment où **il aperçoit un véhicule devant lui**.

# Proposition II (2/2)

## Vérification de la **concordance** entre **limite de visibilité** et **vitesse maximale autorisée**

Ainsi :

- pour un **véhicule positionné** en ***xx*** ;
- supposé roulant à la **vitesse limite** en *xx* ***VLim(xx)*** ;

il s'agit de vérifier que $dFrein(VLim(xx)) \leq VisiLim(xx)$, où :

- $dFrein(VLim(xx))$ est la **distance mimimale de freinage** associée à la **vitesse limite** *VLim(xx)*.
- $VisiLim(xx)$ est la **limite de visibilité** en *xx*.

# Proposition III

Prise en compte des **PMVs** dans l'évaluation de la **concordance** entre **limite de visibilité** et **vitesse maximale autorisée**

## Objectif

**Variante de la proposition II** dans laquelle **les notifications affichées par les PMVs influent sur la vitesse limite** en certains points du tunnel.

Il s'agit :

- d'**associer une vitesse limite à chaque message affiché** par un PMV, en tout point $xx$ où le **panneau est visible**. **Eg :**
  - *"trafic normal"* $\Rightarrow VLim(xx) = 70\ KM/H$
  - *"trafic dense"* $\Rightarrow VLim(xx) = 40\ KM/H$
- d'évaluer la **distance minimale de freinage** dans chaque cas et **vérifier qu'elle concorde avec la limite de visibilité**.

# Proposition IV

**Liveness :** vérification de l'**efficacité des plans de feux**

## Objectif

Vérifier si le **plan de feux** associé à chaque niveau de trafic **contribue à résorber le trafic**.

Spécifier formellement :

- les **véhicules présents** dans le tunnel ;
- les **véhicules entrant** pendant une durée définie ;
- les **véhicules sortant** pendant une **durée d'activation du feu vert** (croisement Nazareth/Wiliam) ;

et **vérifier** si la **durée du feu vert** est **suffisante** pour que la **congestion se résorbe** après un certain intervalle de temps.

# Proposition V

## Écart entre trafic observé et trafic réel

### Objectif

Définir et vérifier des **contraintes sur l'écart entre le trafic observé et le trafic réel**. Eg :

- le niveau de **trafic observé doit être le même** que le niveau de **trafic réel**.
- l'écart entre le **nombre** de véhicules **observés** et le **nombre** de véhicules **réellement présents** doit être **de** $\pm$ **5**.

Quelles contraintes vérifier ?

# D'autres propositions ?