



**HAL**  
open science

## Modélisation hybride dynamique de flux de trafic

Hassane Abouaïssa, Yoann Kubera, Gildas Morvan

► **To cite this version:**

Hassane Abouaïssa, Yoann Kubera, Gildas Morvan. Modélisation hybride dynamique de flux de trafic. 2013. hal-00939129v1

**HAL Id: hal-00939129**

**<https://hal.science/hal-00939129v1>**

Submitted on 30 Jan 2014 (v1), last revised 19 Feb 2015 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Modélisation hybride dynamique de flux de trafic

CISIT Phase 5 — Projet ISART

Rapport scientifique 2013

english version : <http://arxiv.org/abs/1401.6773>

Hassane Abouaïssa, Yoann Kubera, Gildas Morvan

<http://www.lgi2a.univ-artois.fr/~morvan/>  
gildas.morvan@univ-artois.fr

Univ Lille Nord de France, F-59000 Lille, France  
UArtois, LGI2A, F-62400, Béthune, France



**Acknowledgments** This work has been supported by International Campus on Safety and Intermodality in Transportation (CISIT), the Nord-Pas-de-Calais Region, the European Community, the Regional Delegation for Research and Technology, the Ministry of Higher Education and Research, and the National Center for Scientific Research (CNRS).

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Contexte . . . . .	3
1.2	Motivations . . . . .	4
<b>2</b>	<b>SIMILAR</b>	<b>4</b>
2.1	Motivations . . . . .	4
2.2	Travaux connexes . . . . .	5
2.3	Le micro noyau de SIMILAR . . . . .	5
2.4	Architecture de SIMILAR . . . . .	5
2.5	Le moteur de simulation . . . . .	6
2.6	Probes . . . . .	7
<b>3</b>	<b>JAM-FREE</b>	<b>7</b>
3.1	Motivations . . . . .	7
3.2	Travaux connexes . . . . .	8
3.3	Cas d'utilisation de JAM-FREE . . . . .	8
3.4	Structure du réseau routier . . . . .	8
3.5	Comportement des véhicules . . . . .	9
3.5.1	Modèle d'accélération . . . . .	9
3.5.2	Modèle de changement de voie . . . . .	11
3.6	Interfaces utilisateur . . . . .	11
<b>4</b>	<b>Conclusion</b>	<b>12</b>
4.1	Délivrables . . . . .	12
4.2	Perspectives . . . . .	12
	<b>Annexes</b>	<b>12</b>
<b>A</b>	<b>Fichiers XML de JAM-FREE</b>	<b>12</b>
<b>B</b>	<b>JAM-FREE Framework</b>	<b>16</b>
<b>C</b>	<b>SonarQube</b>	<b>23</b>
	<b>Références</b>	<b>23</b>

# Résumé

La simulation du trafic routier sur des réseaux de grande échelle est un problème compliqué car il suppose d'intégrer dans un même modèle différentes approches. Ainsi, les sections autoroutières sont généralement représentées à l'aide de modèles macroscopiques alors que pour les sections urbaines, des modèles microscopiques sont utilisés. De manière générale, les modèles microscopiques sont intéressants lorsque les interactions entre véhicules, ainsi que la topologie du réseau deviennent complexes.

Les modèles intégrant ces différents niveaux de représentation sont généralement qualifiés d'hybrides. Par ailleurs, ils sont généralement "statiques" : à chaque portion du réseau est associée une représentation unique qui ne changera pas au cours de la simulation. Afin de palier cette limitation, nous avons débuté en 2013 dans le cadre du projet ISART le développement d'un simulateur multi-agent multi-niveaux de flux de trafic routier nommé JAM-FREE permettant :

- de simuler des réseaux routier de grande taille efficacement en adaptant dynamiquement le niveau de détail,
- de tester de nouveaux algorithmes de régulation, d'observation et routage.

Ce simulateur repose sur un framework de modélisation et de simulation multi-agents multi-niveaux nommé SIMILAR (SIMulations with Multi-Level Agents and Reactions), implémenté en Java (Morvan et Kubera, 2014b) et distribué prochainement sous licence libre.

Dans ce rapport, nous présentons ces résultats scientifiques ainsi que les publications associées.

## I Introduction

Cet article rapporte les travaux conduits par Hassane Abouaïssa <sup>1</sup>, Yoann Kubera <sup>2</sup> and Gildas Morvan <sup>3</sup> durant la phase 5 de CISIT <sup>4</sup> dans le projet ISART.

### I.1 Contexte

Un flux d'agent mobiles peut être observé à différentes échelles. Ainsi dans la modélisation du flux de trafic routier, trois niveaux sont généralement considérés : *micro*, *méso* et *macro*, représentant respectivement les interactions entre véhicules, groupes de véhicules partageant certaines propriétés (comme une position voisine ou une même destination) et enfin flux de véhicules. Chaque approche est utile dans un contexte particulier : les modèles micro et méso permettent de simuler des réseaux à topologie complexe comme les zones urbaines, alors que les modèles macro permettent de développer des stratégies de contrôle pour prévenir les congestions sur les autoroutes.

Cependant, pour simuler des réseaux routiers de grande taille, il peut être intéressant d'intégrer ces différentes représentations dans un même modèle comme le montre la fig. 1. Certains modèles hybrides, associant représentations micro and macro sont présentés dans le tableau 1.

modèle	micro	macro
Magne <i>et al.</i> (2000)	SITRA-B+	SIMRES
Poschinger <i>et al.</i> (2002)	IDM	Payne
Bourrel et Lesort (2003)		LWR
Mammar et Haj-Salem (2006)	optimal velocity	ARZ
Espié <i>et al.</i> (2006)	ARCHISM	SSMT
El hmam (2006)	generic ABM	LWR, ARZ, Payne

TABLE 1 – Principaux modèles hybrides de flux de trafic, adapté de El hmam (2006, p. 42)

1. <http://www.lgi2a.univ-artois.fr/spip/spip.php?article5&idpersonne=5>

2. <http://www.yoannkubera.net/>

3. <http://www.lgi2a.univ-artois.fr/~morvan/>

4. <http://www.cisit.org/>

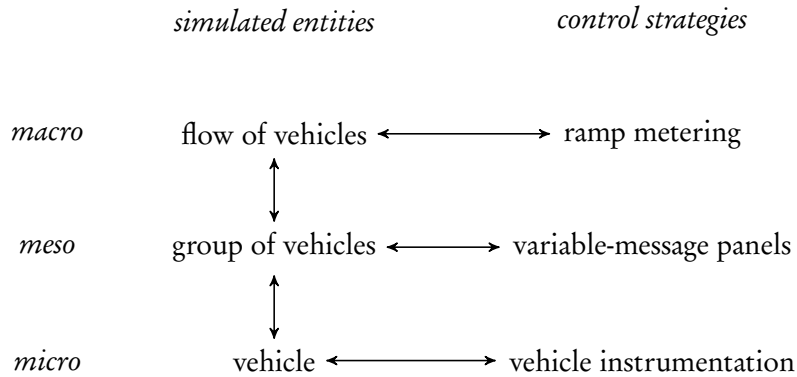


FIGURE 1 – Simulation et contrôle hybride de flux de trafic

## 1.2 Motivations

Les modèles présentés dans le tableau 1 partagent la même limitation : les connexions entre niveaux sont fixés *a priori* et ne peuvent être changées durant la simulation. Pourtant, pour pouvoir observer certains phénomènes émergents comme la formation de congestions ou déterminer précisément la localisation d’une congestion dans une section modélisée par un modèle macro, une approche hybride dynamique est requise (Morvan et Kubera, 2014b; Sewall *et al.*, 2011).

La modélisation multi-agents multi-niveaux est une approche intéressante pour simuler de tels systèmes (Gaud *et al.*, 2008; Gil-Quijano *et al.*, 2010; Gil-Quijano *et al.*, 2012; Picault et Mathieu, 2011; Vo, 2012; Vo *et al.*, 2012a,b).

En effet, ces travaux proposent une large gamme de techniques pour adapter dynamiquement le niveau de détail des simulations, intégrer des modèles hétérogènes ou détecter et réifier des phénomènes émergents (Morvan, 2013).

Ainsi, en 2013, nous avons commencé le développement d’un simulateur multi-agents multi-niveaux appelé JAM-FREE dans le cadre du projet ISART. Il permet de simuler les grands réseaux routiers efficacement en utilisant un niveau de détail dynamique.

Ce simulateur repose sur un framework de modélisation baptisé SIMILAR — appelé anciennement IRM4MLS (Morvan et Jolly, 2012; Morvan *et al.*, 2011; Soyez *et al.*, 2013) — pour **SI**mulations with **Mu**lti-**L**evel **A**gents and **R**eactions.

Les sections suivantes présentent les deux délivrables que nous avons développés : SIMILAR et JAM-FREE.

## 2 SIMILAR

### 2.1 Motivations

La simulation de systèmes complexes nécessite souvent l’intégration de connaissances provenant de différentes sources pour obtenir des résultats pertinents. Ces sources peuvent être des personnes dont les champs d’application diffèrent ou plus simplement de points de vue différents sur le même phénomène.

Pourtant, les méta-modèles classiques pour la simulation multi-agents ne sont pas conçus pour intégrer de tels systèmes : leur représentation des agents, de l’environnement et de la dynamique temporelle du système repose sur un seul point de vue.

Pour répondre à ce problème, nous avons développé une approche générique appelée SIMILAR pour

concevoir des simulations (fig. 2). Cette approche repose une représentation multi-agents multi-niveaux fondée sur le principe influence/réaction afin d'intégrer ces différents points de vue (Ferber et Müller, 1996; Michel, 2007a,b; Morvan et Jolly, 2012; Morvan *et al.*, 2011; Soyez *et al.*, 2013). Cette approche comprend un modèle formel générique, une méthodologie et une API de simulation qui préserve la structure et la sémantique du modèle formel (Morvan et Kubera, 2014b).



FIGURE 2 – Logo de SIMILAR

## 2.2 Travaux connexes

De nombreux méta-modèles dédiés à la simulation multi-agents multi-niveaux ont été proposés dans la littérature. Une étude complète peut être trouvée dans Morvan (2013). On notera que ces travaux imposent généralement des contraintes fortes sur les modèles (par exemple sur la structure des graphes d'interaction ou d'influence). À l'opposé, SIMILAR permet de simuler tout type de modèle multi-niveaux car il repose sur un modèle formel d'interaction (Ferber et Müller, 1996).

## 2.3 Le micro noyau de SIMILAR

SIMILAR est basé sur le concept de micro noyau, qui définit les classes principales permettant d'effectuer des simulations. Il fournit une correspondance directe entre les concepts théoriques et l'implémentation concrète d'une simulation. Il ne fournit que l'implémentation minimale de ces concepts, laissant ainsi au modélisateur une grande liberté. Ce noyau est par ailleurs complété par un noyau étendu, fournissant toute une classe d'outils facilitant la conception, la modification et l'extension de simulations.

Le micro noyau de SIMILAR est constitué de trois composants (fig. 3) :

- L'API du micro noyau, contenant les classes Java de ce noyau.
- Les bibliothèques standard de ce noyau minimal, contenant des implémentations génériques des algorithmes de simulation et des outils d'export des résultats.
- Les exemples illustrant l'utilisation du micro noyau et des bibliothèques communes pour concevoir des simulations.

## 2.4 Architecture de SIMILAR

SIMILAR est conçu pour avoir une architecture hautement personnalisable, tout en rendant la structure des simulations aussi explicite que possible.

Chaque concept de SIMILAR est intégré dans une classe concrète. À haut niveau, cela se traduit par la séparation entre les éléments constitutifs d'une simulation (fig. 4) :

- Le modèle de simulation (*i.e.* "simulation model"), contenant la partie déclarative de la simulation (la description du phénomène simulé);

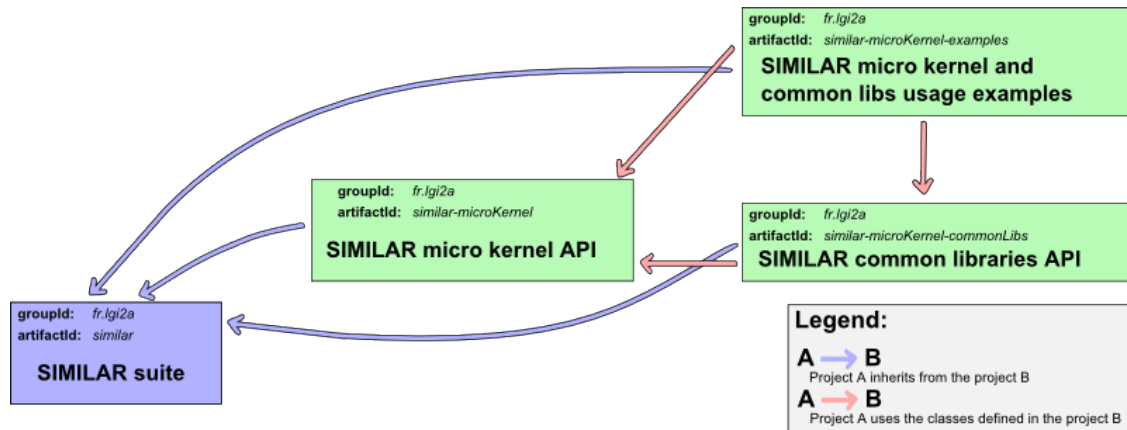


FIGURE 3 – Structure du micro noyau de SIMILAR

- Le moteur de simulation (*i.e.* "simulation engine"), contenant la partie procédurale de la simulation (algorithmes génériques permettant d'exécuter une simulation) ;
- Les outils d'observation (*i.e.* "observation probes"), capables de récupérer des informations sur la simulation et de les exporter dans divers formats.

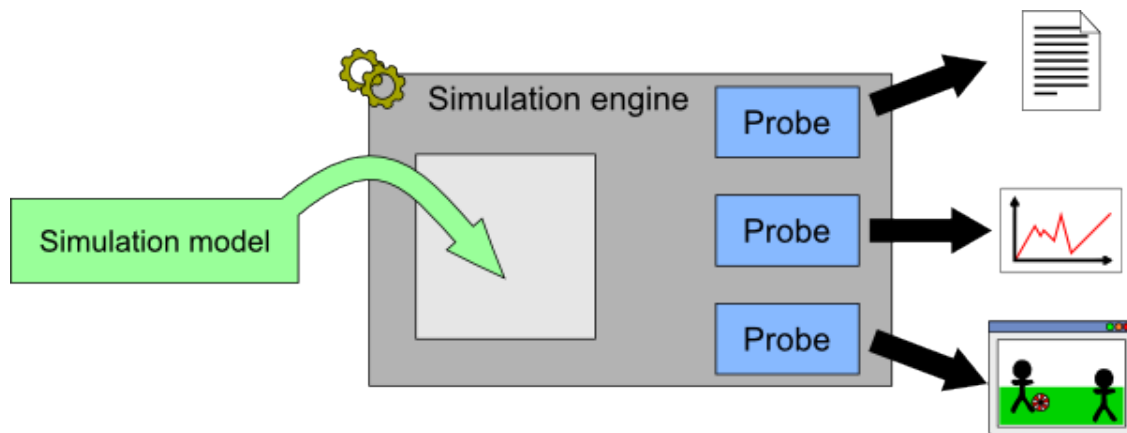


FIGURE 4 – Architecture de SIMILAR

## 2.5 Le moteur de simulation

Dans SIMILAR, la notion de modèle de simulation est séparée de la notion de moteur de simulation pour distinguer les connaissances déclarative et procédurale de la simulation :

- Le modèle de simulation contient la définition des niveaux, les agents, l'action naturelle de l'environnement et la réaction de chaque niveau. Cette connaissance est spécifique au domaine .
- Un moteur de simulation assemble les informations liées à l'exécution d'une simulation-. Il définit l'évolution du temps, quand demander aux agents de percevoir, de mémoriser ou décider, et détermine le moment où l'environnement produit son action naturelle ou encore le moment où la réaction est effectuée. Il est également responsable de l'observation et de l'exportation des données de simulation, ainsi que de la réaction aux influences système (création d'un nouvel agent,

changement de niveau, etc.).

La séparation entre le modèle et le moteur facilite l'optimisation des simulations : le mode d'exécution le plus approprié peut être choisi en fonction de la simulation. En effet, un moteur de simulation peut être implémenté à l'aide de différents mécanismes internes pour gérer l'exécution d'une simulation.

Le moteur de simulation joue des rôles différents dans la simulation :

- Faire évoluer le temps de façon appropriée et gérer :
  - les phases de perception, mémorisation, décision des agents ;
  - l'influence naturelle de l'environnement ;
  - la phase de réaction des niveaux ;
- Assurer que les contraintes liées au modèle temporel restent valides durant toute l'exécution de la simulation. Morvan *et al.* (2011) décrit l'ensemble de ces contraintes ;
- Assurer que les probes d'observation sont mis à jour de façon appropriée ;
- Fournir une réaction aux influences système pour n'importe quel simulation.

Considérer le moteur de simulation comme une abstraction de haut niveau permet facilement de simuler un même modèle sur différentes architectures de calcul comme les CPUs multi-coeurs ou les GPGPU (general purpose graphical processing unit) comme le montre la fig. 5.

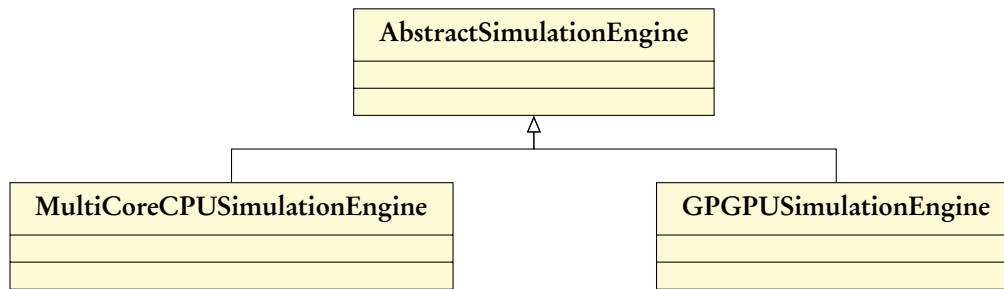


FIGURE 5 – Moteurs de simulation de SIMILAR

## 2.6 Probes

L'observation des données de la simulation est géré au moyen de probes. Les probes sont des objets écoutant l'évolution de l'état de la simulation. Comme l'état de la simulation n'est pas toujours cohérente (en particulier lors du calcul de la réaction), les probes ne décident pas par eux-mêmes quand observer la simulation. Ainsi, ils sont liés à l'état du moteur de simulation. Le moteur de simulation est responsable de dire aux probes lorsque l'état de la simulation est cohérent, et donc quand les probes peuvent observer la simulation et exporter des résultats.

## 3 JAM-FREE

Dans cette section, nous présentons le modèle hybride dynamique de flux de trafic appelé JAM-FREE (Java, Agent and Multi-level based Framework for Road-traffic Examination and Enhancement), basé sur SIMILAR.

### 3.1 Motivations

Nos motivations pour développer JAM-FREE sont :

1. simuler le trafic routier sur de très grandes zones comprenant à la fois les villes et les autoroutes,



2. être en mesure de tester différentes stratégies de régulation ou de redirection du trafic,
3. évaluer leur influence précise sur la fluidité de la circulation, et
4. comprendre pourquoi des perturbations de la circulation se produisent.

Le premier objectif est facilement atteint en utilisant des modèles macroscopiques au détriment du quatrième but. A l'inverse, le quatrième objectif est facilement atteint en utilisant des modèles microscopiques au détriment du premier objectif. Pour faire face à ce paradoxe, nous avons conçu un modèle hybride dynamique permettant de bénéficier des deux approches.

Les avantages de cette approche hybride comprennent la capacité :

- d'obtenir des informations quantitatives et qualitatives sur la circulation routière, en utilisant des représentations de simulation respectivement macroscopiques et microscopiques dans la même simulation.
- de basculer entre ces représentations localement :
  - Pour les besoins de la simulation. Par exemple comprendre la source d'un embouteillage.
  - Pour gérer les contraintes de calcul. Par exemple la gestion de la charge du processeur.
- expérimenter des stratégies de routage macroscopiques et microscopiques, i.e., des stratégies d'équilibrage de la charge sur différentes routes.

### 3.2 Travaux connexes

A notre connaissance, le seul travail décrivant un modèle de ce type est (Sewall *et al.*, 2011)<sup>5</sup>. Cependant, les auteurs mettent l'accent sur la visualisation de données plutôt que la simulation précise des flux de trafic.

### 3.3 Cas d'utilisation de JAM-FREE

Dans le cadre de ce projet, nous avons défini une dizaine de cas d'utilisation pouvant être traités uniquement par des modèles hybrides dynamiques, et donc par JAM-FREE. Dans cette section, nous en présentons très brièvement deux.

Le modèle hybride est capable de changer dynamiquement la représentation de la circulation sur une partie du réseau routier. Cette fonction peut être utilisée pour passer :

- d'une représentation microscopique à une représentation macroscopique lorsque le CPU est surchargé, dans le but de réduire sa charge ;
- d'une représentation macroscopique à une représentation microscopique pour connaître la raison pour laquelle un embouteillage est apparu.

### 3.4 Structure du réseau routier

Dans JAM-FREE, le réseau routier peut être divisé dynamiquement en sous-ensembles que nous nommons "clusters" (voir fig. 6). Le trafic routier est simulé sur chaque cluster à l'aide de modèles hétérogènes dépendant du contexte. Ces derniers sont soit des modèles microscopiques, soit des modèles macroscopiques (voir fig. 7).

Pour des raisons d'efficacité, nous choisissons de ne pas modéliser le réseau routier à un niveau physique. En effet, un tel modèle nécessite d'intégrer aux véhicules des perceptions complexes. Ces calculs peuvent être évités, puisque dans notre cas d'utilisation les conducteurs sont généralement rationnels et conduisent en respectant les voies signalées au sol. Ainsi, le réseau routier n'est modélisé qu'à un niveau sémantique.

---

<sup>5</sup>. Les auteurs maintiennent une page web liée à leur article : [http://gamma.cs.unc.edu/HYBRID\\_TRAFFIC/](http://gamma.cs.unc.edu/HYBRID_TRAFFIC/)

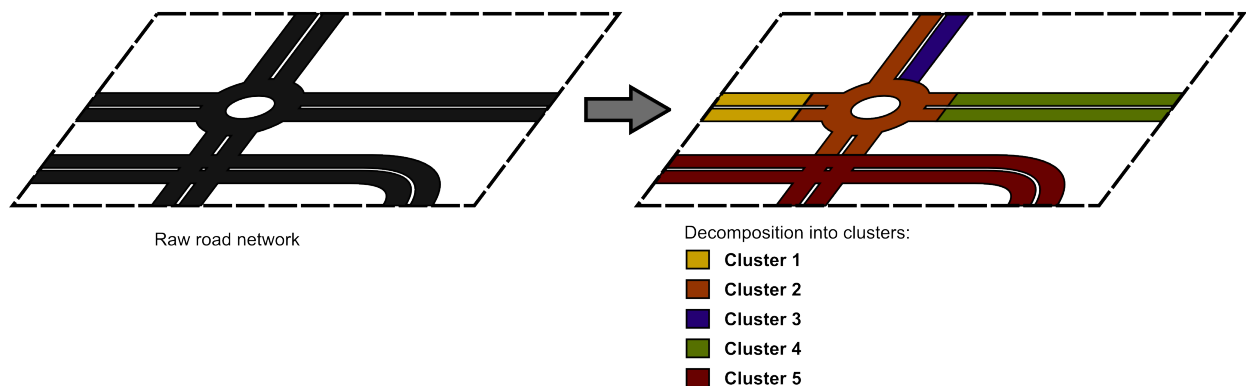


FIGURE 6 – SIMILAR road network structure

Le réseau routier est modélisé comme un ensemble de routes interconnectées. Chaque route comporte un ensemble de voies (généralement de 1 à 5). Les routes contiennent également la signalisation verticale (par exemple un panneau d'arrêt ou de limitation de vitesse), liée à toutes les voies ou à des voies spécifiques (par exemple une voie de sortie ou une voie dédiée aux véhicules lents). Dans notre modèle, nous intégrons un sous-ensemble des panneaux de signalisation français, pertinents pour notre approche.

Les points de connexion entre des routes sont appelés noeuds. Nous distinguons différents types de noeuds :

- croisements,
- ronds-points,
- voies d'insertion,
- voies d'extraction.

La dénomination et l'identification des composants de réseau routier suivent les spécifications de la Sétra<sup>6 7</sup> (Sétra, 2010).

### 3.5 Comportement des véhicules

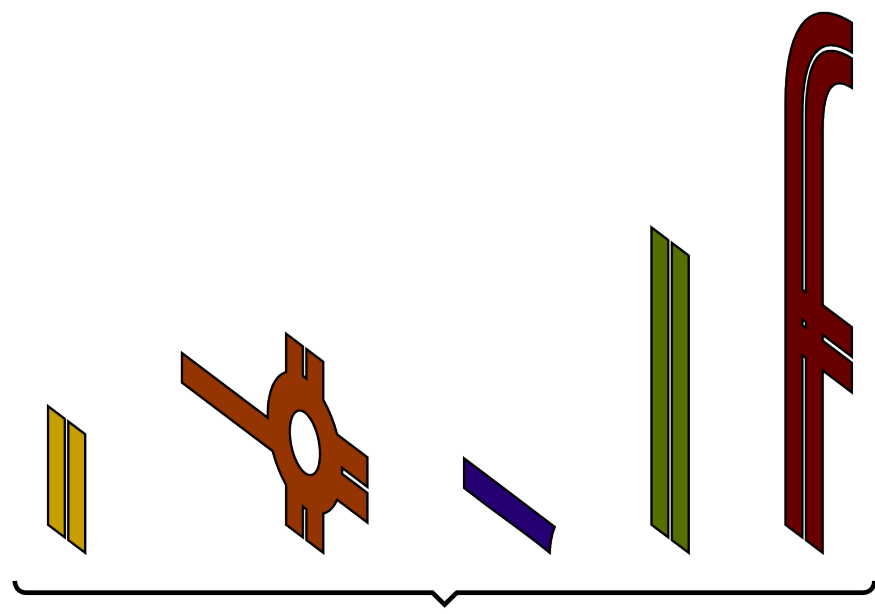
Le comportement d'un véhicule dans notre simulation est conçu pour être modulaire, personnalisable et facilement extensible. Pour atteindre ces propriétés, le comportement d'un véhicule repose sur le patron de conception "chaîne de responsabilité".

Pour simplifier la conception du comportement des véhicules, le comportement est conçu comme la somme de trois parties distinctes (voir fig. 8). Le concepteur n'est toutefois pas restreint à cette structure, et peut la complexifier si besoin en est (par exemple en spécifiant un comportement en ville et un comportement hors ville ou tout autre comportement dont la nature change en fonction des circonstances) :

- Un comportement de navigation quand la voie actuelle du véhicule ne conduit pas à sa destination. ce comportement consiste à changer la voie du véhicule jusqu'à ce que la voie actuelle du véhicule conduise à la destination désirée.
- Un comportement de dépassement.
- Un comportement quand aucun changement de voie n'est requis (modèle d'accélération).

6. <http://www.setra.developpement-durable.gouv.fr>

7. <http://dtrf.setra.fr/pdf/pj/Dtrf/0005/Dtrf-0005792/DT5792.pdf>

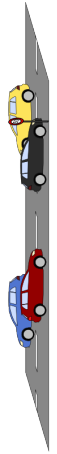


**Cluster 1:** simulation with macroscopic traffic flow model

$$\frac{(\partial \rho)}{(\partial t)} + \frac{(\partial q)}{(\partial x)} = 0 \quad (1)$$

$$\frac{(\partial v)}{(\partial t)} + v \frac{(\partial \rho)}{(\partial x)} = \frac{(V(\rho) - v)}{(\tau)} - \frac{C_0^2(\partial \rho)}{\rho (\partial x)} + \frac{f(\partial^2 v)}{\rho (\partial x^2)} \quad (2)$$

**Cluster 2:** simulation with microscopic multi-agent system



**Cluster 3:** simulation with macroscopic traffic flow model

$$\frac{(\partial \rho)}{(\partial t)} + \frac{(\partial q)}{(\partial x)} = 0 \quad (1)$$

$$\frac{(\partial v)}{(\partial t)} + v \frac{(\partial \rho)}{(\partial x)} = \frac{(V(\rho) - v)}{(\tau)} - \frac{C_0^2(\partial \rho)}{\rho (\partial x)} + \frac{f(\partial^2 v)}{\rho (\partial x^2)} \quad (2)$$

**Cluster 4:** simulation with macroscopic traffic flow model

$$\frac{(\partial \rho)}{(\partial t)} + \frac{(\partial q)}{(\partial x)} = 0 \quad (1)$$

$$\frac{(\partial v)}{(\partial t)} + v \frac{(\partial \rho)}{(\partial x)} = \frac{(V(\rho) - v)}{(\tau)} - \frac{C_0^2(\partial \rho)}{\rho (\partial x)} + \frac{f(\partial^2 v)}{\rho (\partial x^2)} \quad (2)$$

**Cluster 5:** simulation with microscopic traffic flow model

$$\dot{x}_n(t) = v_n(t) = F(v_n(t), s_n(t), v_{n-1}(t))$$

$$\dot{v}_n(t) = f(x_n(t), v_n(t), x_{n-1}(t), v_{n-1}(t), \dots, x_{n-ma}(t), v_{n-ma}(t))$$

$$e_n^{t+1} = f(e_n^t, v_n^t, v_{n-1}^t, \dots)$$

FIGURE 7 – SIMILAR traffic representations

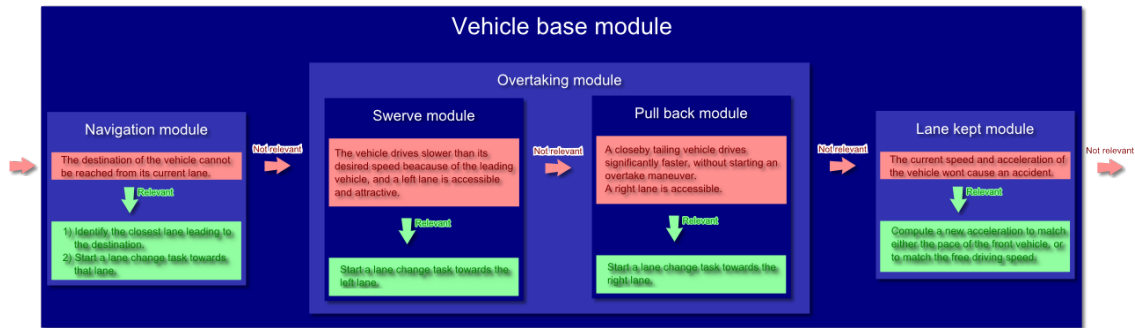


FIGURE 8 – Comportements des véhicules dans SIMILAR

### 3.5.1 Modèle d'accélération

JAM-FREE implémente le modèle IDM (Intelligent-Driver Model) pour simuler le comportement d'accélération des véhicules (Kesting *et al.*, 2010).

IDM est un modèle de flux de trafic microscopique, i.e., chaque combinaison véhicule-conducteur constitue un agent dans la simulation. Ce modèle caractérise l'état de la circulation à un moment donné par les positions et les vitesses de tous les véhicules simulés. En cas de trafic à plusieurs voies, l'indice de la voie complète la description de l'état. IDM est un modèle de poursuite. Dans ces modèles, la décision d'un conducteur d'accélérer ou de freiner ne dépend que de sa propre vitesse et de la position et de la vitesse du véhicule situé immédiatement en avant. Un changement de voie, cependant, dépendra de tous les véhicules voisins (voir le modèle de changement de voie MOBIL).

### 3.5.2 Modèle de changement de voie

Les changements de voies ont lieu si :

- une autre voie est plus attrayante (critère d'incitation),
- et la modification peut être effectuée en toute sécurité (critère de sécurité).

Dans JAM-FREE, le modèle de changement de voie utilisé est une version très fortement inspirée du modèle de changement de voie MOBIL <sup>8</sup>.

Ces différents modèles sont décrits précisément dans la documentation de JAM-FREE.

## 3.6 Interfaces utilisateur

Des modèles et simulations JAM-FREE peuvent être facilement créés à l'aide d'un système hybride XML/Java :

- les simulations utilisant les modèles comportementaux prédéfinis (IDM et MOBIL modifiés) peuvent être créées en utilisant seulement des fichiers de paramétrage au format XML,
- les simulations plus complexes peuvent être créées en deux étapes : 1) la création de classes Java décrivant les nouveaux comportements et 2) la mention de ces nouvelles classes dans les fichiers de paramétrage au format XML.

Afin de faciliter la conception de simulations portant sur un même réseau routier, un modèle comprend plusieurs fichiers XML :

8. <http://xxx.uni-augsburg.de/abs/cond-mat/0002177>

- un fichier décrivant le réseau routier ;
- un fichier identifiant la position des points de génération et de destruction du trafic routier ;
- Des fichiers décrivant les paramètres utilisés pour chacun de ces points.

Un exemple de simulation est illustré à l'annexe A.

Pour gérer simplement des simulations, une interface utilisateur graphique appelée JFF (JAM-FREE Framework) a également été développée (voir l'annexe B).

## 4 Conclusion

Dans le cadre du projet ISART de CISIT nous développons un simulateur hybride dynamique de flux de trafic appelé JAM-FREE, reposant sur le méta-modèle SIMILAR. Ces logiciels sont développés selon les meilleures pratiques de génie logiciel et à l'aide de nombreux outils comme SonarQube<sup>9</sup> (voir l'annexe C), Maven<sup>10</sup>, Jenkins<sup>11</sup> ou Hamcrest<sup>12</sup>. Des tests unitaires exhaustifs ont été effectués pour vérifier les moteurs de simulation et les modèles. Par exemple, 545800 tests (265Mo de code source), chacun décrivant un cas d'utilisation spécifique, ont été générés pour valider le modèle de perception de JAM-FREE.

### 4.1 Délivrables

Le méta-modèle et le moteur de simulation de SIMILAR ont été implémentés en Java, et seront prochainement disponibles librement sous licence CeCILL-B<sup>13</sup>. Le projet est pour le moment hébergé sur <https://forge.univ-artois.fr> avec un accès restreint au code.

JAM-FREE est encore en développement. Cependant, des démonstrations préliminaires peuvent être distribuées aux partenaires de CISIT si nécessaire.

Deux articles sont en préparation :

- Morvan et Kubera (2014b), décrivant le méta-modèle SIMILAR,
- Morvan et Kubera (2014a), proposant de nouvelles façons de gérer l'incertitude dans les simulations multi-agents en utilisant la théorie des fonctions de croyance.

### 4.2 Perspectives

Durant la prochaine phase de CISIT, nous allons poursuivre le développement de JAM-FREE. Ainsi l'un de nos objectifs est d'intégrer les données issues du système d'information géographique open-source OpenStreetMap<sup>14</sup>.

# Appendices

## A Fichiers XML de JAM-FREE

Une simulation JAM-FREE minimale est définie par un ensemble de fichiers XML. Un exemple simple est donné dans cette section. Le fichier principal spécifie les options principales de simulation ainsi

---

9. <http://www.sonarqube.org>

10. <http://maven.apache.org>

11. [jenkins-ci.org](http://jenkins-ci.org)

12. <https://code.google.com/p/hamcrest/>

13. [http://cecill.info/licences/Licence\\_CeCILL-B\\_V1-en.html](http://cecill.info/licences/Licence_CeCILL-B_V1-en.html)

14. <http://www.openstreetmap.org>

que les fichiers XML décrivant la structure du réseau routier et la génération de trafic.

### Listing 1 – JAM-FREE : main XML file

```
<?xml version="1.0" encoding="UTF-8"?>
<jamFreeModel>

  <!-- Identify the file where the infrastructure of the road network is described. -->
  <roadNetwork fileName="infrastructure/navigation-infrastructure.xml" />

  <!-- Describe how time moves through simulation -->
  <time sampling="0.05"> <!-- A time step occurs every 0.05 second of the simulated world (optional). -->
    <simulationEnd time="180" /> <!-- The simulation runs during 60 seconds. -->
    <microscopicTrafficEvolution period="1" /> <!-- The "microscopic - traffic" level acts at each time step (optional) -->
  </time>

  <!-- Define the initial state of the different levels of the simulation. -->
  <levels>
    <!-- Identify the file where the initial state of the "microscopic - traffic" level is described. -->
    <microscopicTrafficLevel fileName="microscopicLevel/microscopicLevel-example1.xml" />
  </levels>

  <!-- Define how the results of the simulation can be seen. -->
  <probes />
</jamFreeModel>
```

Les fichiers d'infrastructure décrivent les composants de le route contenus dans le réseau routier et leurs connexions.

### Listing 2 – JAM-FREE : infrastructure XML file

```
<?xml version="1.0" encoding="UTF-8"?>
<roadnetwork>
<!--
-->
  Description of the road components contained in the road network.
-->
  <road id="N9PR1_section1" numberOfLanes="2" width="4" semanticLength="1000">
    <type value="national"/>
    <input plo="N9PR1" distanceFromPlo="0">
      <laneIdRange from="0" to="1"/>
    </input>
    <output plo="N9PR1" distanceFromPlo="1000">
      <laneIdRange from="0" to="1"/>
    </output>
    <shape>
      <point x="0" y="8"/>
      <point x="1000" y="8"/>
    </shape>
  </road>
  <road id="N9PR1_section2" numberOfLanes="2" width="4" semanticLength="500">
    <type value="national"/>
    <input plo="N9PR1" distanceFromPlo="1000">
      <laneIdRange from="0" to="1"/>
    </input>
    <output plo="N9PR1" distanceFromPlo="1500">
      <laneIdRange from="0" to="1"/>
    </output>
    <reachableDestinations>
      <laneDestination>
        <laneId id="1" />
        <destination name="Paris" />
      </laneDestination>
      <laneDestination>
        <laneId id="0" />
        <destination name="Dunkerque" />
        <destination name="Arras" />
      </laneDestination>
    </reachableDestinations>
    <shape>
      <point x="1000" y="8"/>
      <point x="1500" y="8"/>
    </shape>
  </road>
  <road id="D19PR1_section1" numberOfLanes="1" width="4" semanticLength="300">
    <type value="national"/>
    <input plo="N9PR1" distanceFromPlo="1500">
      <laneId id="1"/>
    </input>
    <output plo="D19PR1" distanceFromPlo="300">
      <laneId id="0"/>
    </output>
    <reachableDestinations>
      <laneDestination>
        <laneId id="1" />
        <destination name="Paris" />
      </laneDestination>
    </reachableDestinations>
    <shape>
```

```

        <point x="1500" y="4"/>
        <point x="1800" y="4"/>
    </shape>
</road>
<road id="D20PR1_section1" numberOfLanes="1" width="4" semanticLength="300">
    <type value="national"/>
    <input plo="N9PR1" distanceFromPlo="1500">
        <laneId id="0"/>
    </input>
    <output plo="D20PR1" distanceFromPlo="300">
        <laneId id="0"/>
    </output>
    <reachableDestinations>
        <laneDestination>
            <laneId id="0" />
            <destination name="Dunkerque" />
            <destination name="Arras" />
        </laneDestination>
    </reachableDestinations>
    <shape>
        <point x="1500" y="8"/>
        <point x="1501" y="8"/>
        <point x="1651" y="158"/>
    </shape>
</road>
<!--
-->
    Description of the connection between the road components.
-->
    <connection outputComponentId="N9PR1_section1" inputComponentId="N9PR1_section2">
        <input plo="N9PR1" distanceFromPlo="1000">
            <laneIdRange from="0" to="1"/>
        </input>
        <output plo="N9PR1" distanceFromPlo="1000">
            <laneIdRange from="0" to="1"/>
        </output>
    </connection>
    <connection outputComponentId="N9PR1_section2" inputComponentId="D19PR1_section1">
        <input plo="N9PR1" distanceFromPlo="1500">
            <laneId id="1"/>
        </input>
        <output plo="N9PR1" distanceFromPlo="1500">
            <laneId id="1"/>
        </output>
    </connection>
    <connection outputComponentId="N9PR1_section2" inputComponentId="D20PR1_section1">
        <input plo="N9PR1" distanceFromPlo="1500">
            <laneId id="0"/>
        </input>
        <output plo="N9PR1" distanceFromPlo="1500">
            <laneId id="0"/>
        </output>
    </connection>
    <road id="D19PR1_section2" numberOfLanes="2" width="4" semanticLength="500">
        <type value="national"/>
        <input plo="D19PR1" distanceFromPlo="300">
            <laneIdRange from="0" to="1" />
        </input>
        <output plo="D19PR1" distanceFromPlo="800">
            <laneIdRange from="0" to="1" />
        </output>
        <explicitSpecification>
            <!-- We tell that the lane having input point "0" starts at the beginning of the road. -->
            <laneStartingPoint inputId="0" />
        </explicitSpecification>
        <shape>
            <point x="1800" y="8"/>
            <point x="2300" y="8"/>
        </shape>
    </road>
    <road id="D19PR1_section3" numberOfLanes="2" width="4" semanticLength="500">
        <type value="national"/>
        <input plo="D19PR1" distanceFromPlo="800">
            <laneIdRange from="0" to="1" />
        </input>
        <output plo="D19PR1" distanceFromPlo="1300">
            <laneIdRange from="0" to="1" />
        </output>
        <reachableDestinations>
            <laneDestination>
                <laneId id="1" />
                <destination name="Paris" />
            </laneDestination>
            <laneDestination>
                <laneId id="0" />
                <destination name="Lille" />
            </laneDestination>
        </reachableDestinations>
        <shape>
            <point x="2300" y="8"/>
            <point x="2800" y="8"/>
        </shape>
    </road>

```

```

    </shape>
  </road>
  <connection outputComponentId="D19PR1_section1" inputComponentId="D19PR1_section2">
    <input plo="D19PR1" distanceFromPlo="300">
      <laneId id="1"/>
    </input>
    <output plo="D19PR1" distanceFromPlo="300">
      <laneId id="0"/>
    </output>
  </connection>
  <connection outputComponentId="D19PR1_section2" inputComponentId="D19PR1_section3">
    <input plo="D19PR1" distanceFromPlo="800">
      <laneIdRange from="0" to="1"/>
    </input>
    <output plo="D19PR1" distanceFromPlo="800">
      <laneIdRange from="0" to="1"/>
    </output>
  </connection>

  <road id="D25PR1_section1" numberOfLanes="1" width="4" semanticLength="300">
    <type value="national"/>
    <input plo="D19PR1" distanceFromPlo="1300">
      <laneId id="0"/>
    </input>
    <output plo="D25PR1" distanceFromPlo="300">
      <laneId id="0"/>
    </output>
    <reachableDestinations>
      <laneDestination>
        <laneId id="0" />
        <destination name="Lille" />
      </laneDestination>
    </reachableDestinations>
    <shape>
      <point x="2800" y="8"/>
      <point x="2802" y="8"/>
      <point x="2802" y="306"/>
    </shape>
  </road>
  <connection outputComponentId="D19PR1_section3" inputComponentId="D25PR1_section1">
    <input plo="D19PR1" distanceFromPlo="1300">
      <laneId id="0"/>
    </input>
    <output plo="D19PR1" distanceFromPlo="1300">
      <laneId id="0"/>
    </output>
  </connection>
  <road id="D26PR1_section1" numberOfLanes="1" width="4" semanticLength="300">
    <type value="national"/>
    <input plo="D19PR1" distanceFromPlo="1300">
      <laneId id="1"/>
    </input>
    <output plo="D26PR1" distanceFromPlo="300">
      <laneId id="0"/>
    </output>
    <reachableDestinations>
      <laneDestination>
        <laneId id="1" />
        <destination name="Paris" />
      </laneDestination>
    </reachableDestinations>
    <shape>
      <point x="2800" y="4"/>
      <point x="3100" y="4"/>
    </shape>
  </road>
  <connection outputComponentId="D19PR1_section3" inputComponentId="D26PR1_section1">
    <input plo="D19PR1" distanceFromPlo="1300">
      <laneId id="1"/>
    </input>
    <output plo="D19PR1" distanceFromPlo="1300">
      <laneId id="1"/>
    </output>
  </connection>
</roadnetwork>

```

Les fichiers de niveau microscopique définissent des points de génération de véhicule et de destruction.

### Listing 3 – JAM-FREE : microscopic level XML file

```

<?xml version="1.0" encoding="UTF-8"?>
<microTrafficLevel>
<!--
--> Create the generation points at the beginning of the simulated A1 highway (at the PLO A1PR35D)
-->
  <trafficGenerationPoint>
    <location>
      <position roadId="N9PR1_section1" plo="N9PR1" distanceFromPlo="0" laneId="0" />
    </location>
  </trafficGenerationPoint>

```



```

        <behavior>
            <rhythmModel
                javaClass="fr.lgi2a.jamFree.libraries.microscopicLevel.generationRythmModels.FlowMassBased_RhythmModel"
                parametersFile="microscopicLevel/generationPoints/A1PR35D_section1-rhythmParameters.xml"
            />
            <vehicleGenerationModel
                javaClass="fr.lgi2a.jamFree.libraries.microscopicLevel.vehicleGenerationModels.
                    UniformWithIDMForwardAcceleration_WithMobilLaneChange_SpecificDestination_GenerationModel"
                parametersFile="microscopicLevel/generationPoints/A1PR35D_section1-generationParameters.xml"
            />
        </behavior>
    </trafficGenerationPoint>
<!--
-->
    Create the destruction point at the end of the simulated A1 national road (at the PLO A1PR38D)
    -->
    <trafficDestructionPoint>
        <location>
            <position roadId="D20PR1_section1" plo="D20PR1" distanceFromPlo="300" laneId="0" />
        </location>
    </trafficDestructionPoint>
    <trafficDestructionPoint>
        <location>
            <position roadId="D26PR1_section1" plo="D26PR1" distanceFromPlo="300" laneId="0" />
        </location>
    </trafficDestructionPoint>
    <trafficDestructionPoint>
        <location>
            <position roadId="D25PR1_section1" plo="D25PR1" distanceFromPlo="300" laneId="0" />
        </location>
    </trafficDestructionPoint>
</microTrafficLevel>

```

Enfin, chaque point de génération se caractérise par deux fichiers XML décrivant les modalités (paramètres de génération et de rythme) de création des véhicules.

#### Listing 4 – JAM-FREE : generation parameters XML file

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
    <entry key="headwayTimeUsedInGenerationValidation">1.5</entry>
    <entry key="vehicleMinimalGapUsedInGenerationValidation">2.0</entry>

    <entry key="vehicleLength">4.13</entry>
    <entry key="accelerationModel_initialSpeed">20.0</entry>

    <entry key="accelerationModel_maximalAcceleration">1.962</entry>
    <entry key="accelerationModel_maximalDeceleration">5.886</entry>
    <entry key="accelerationModel_maximalSpeed">36.1111</entry>
    <entry key="accelerationModel_minimalVehicleGap">2.0</entry>
    <entry key="accelerationModel_headwayTime">1.5</entry>

    <entry key="lanceChange_accelerationAnticipationModel_maximalAcceleration">0.2</entry>
    <entry key="lanceChange_accelerationAnticipationModel_maximalDeceleration">3.0</entry>
    <entry key="lanceChange_accelerationAnticipationModel_headwayTime">1.4</entry>
    <entry key="lanceChange_accelerationAnticipationModel_minimalVehicleGap">2.5</entry>

    <entry key="lanceChange_manueverDuration">2.5</entry>
    <entry key="lanceChange_maximumSafeDeceleration">4.0</entry>
    <entry key="lanceChange_politenessFactor">0.35</entry>
    <entry key="lanceChange_accelerationThreshold">0.2</entry>
    <entry key="lanceChange_rightLaneAccelerationBias">0.1</entry>

    <entry key="lanceChange_thresholdDistanceComputation_headwayTime">1.5</entry>
    <entry key="lanceChange_thresholdDistanceComputation_minimalVehicleGap">2.0</entry>

    <entry key="checkPoints">Paris</entry>
</properties>

```

#### Listing 5 – JAM-FREE : generation rhythm XML file

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
    <entry key="flowMass">0.20</entry>
</properties>

```

## B JAM-FREE Framework

JAM-FREE Framework (JFF) est une interface graphique qui permet aux utilisateurs de gérer simplement des simulations. Des captures d'écran de JFF sont présentées dans les fig. 9–14.

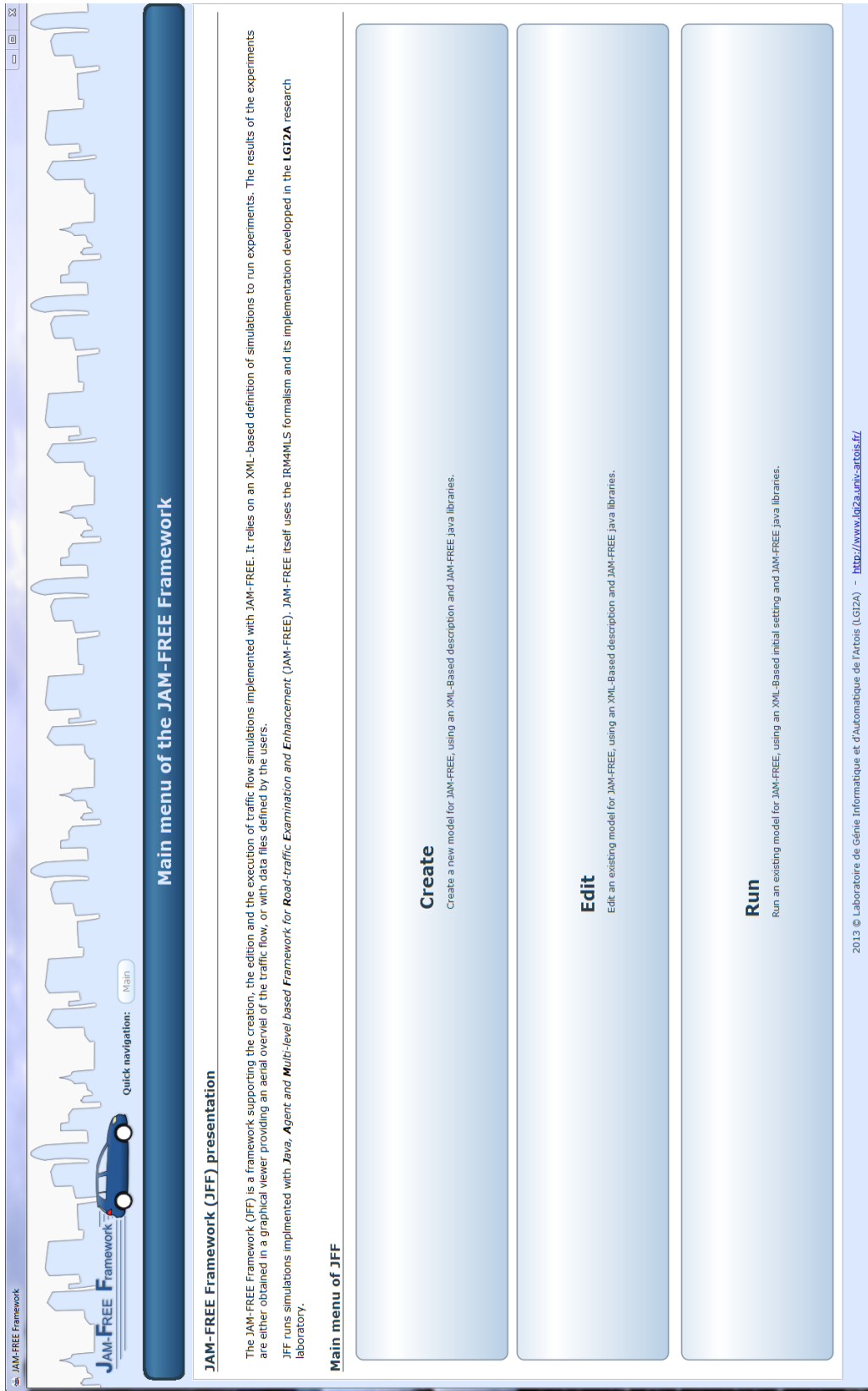


FIGURE 9 – Screenshot de JFF : menu principal

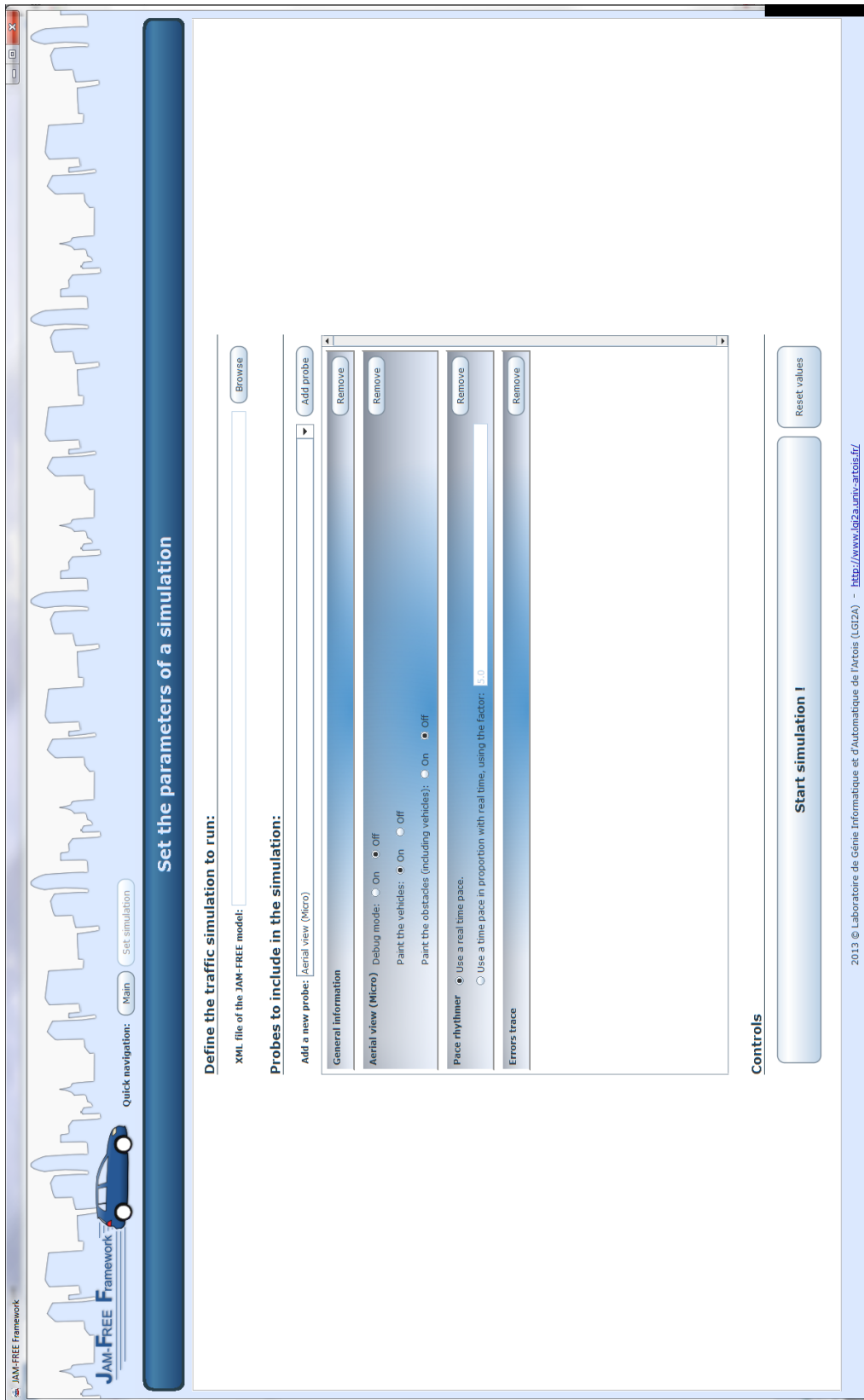


FIGURE 10 – Screenshot de JFF : définir les paramètres d’une simulation

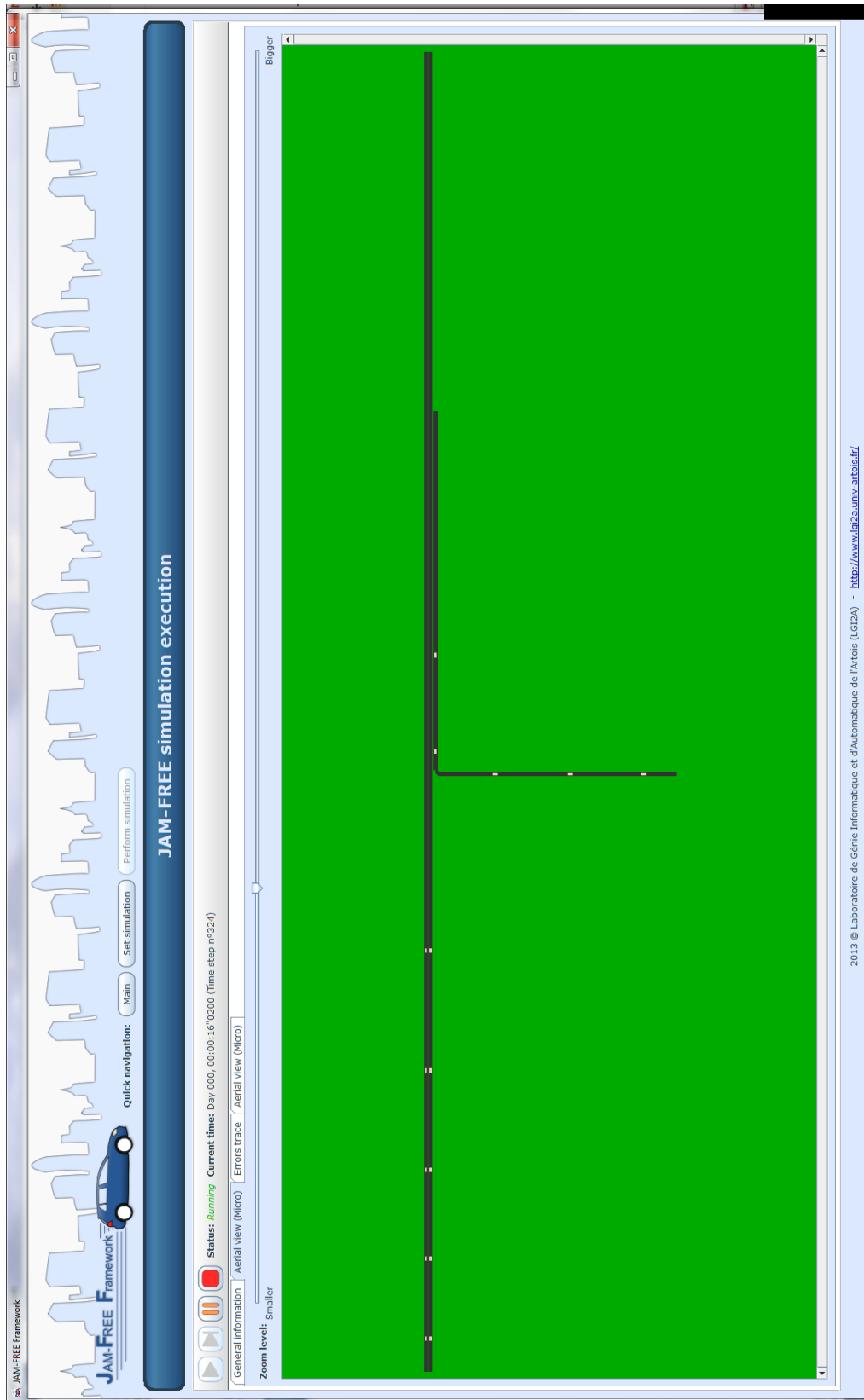


FIGURE 11 – Screenshot de JFF : exécution d'une simulation


[Dashboards](#) [Projects](#) [Measures](#) [Issues](#) [Quality Profiles](#) Yoann KUBERA [Search](#)

---

**Home**

**TOOLS**

Dependencies  
Compare



**Welcome to SonarQube Dashboard**

Since you are able to read this, it means that you have successfully started your SonarQube server. Well done!  
If you have not removed this text, it also means that you have not yet played much with SonarQube. So here are a few pointers for your next step.

- » Do you now want to run analysis on a project?
- » Maybe start customizing dashboards?
- » Or simply browse the [complete documentation](#)?
- » If you have a question or an issue, please visit the [Get Support](#) page.

**My Favourites**

A Name	Last Analysis
No data	

**Projects**

A Name	Version	LOCs	RCJ	Last Analysis
SIMILAR API suite	0.0.1	6 784	96,9% <span style="color: green;">■</span>	16:42

1 results

**Projects**

Size: Lines of code    Color: Rules compliance 0.0%  100.0%

SIMILAR API suite

SonarQube™ technology is powered by SonarSource SA  
Version 3.7.3 - Community - Documentation - Get Support - Plugins

FIGURE 12 – Screenshot de SonarQube : Ecran principal montrant que SIMILAR est conforme à 96.9% aux meilleures pratiques de développement

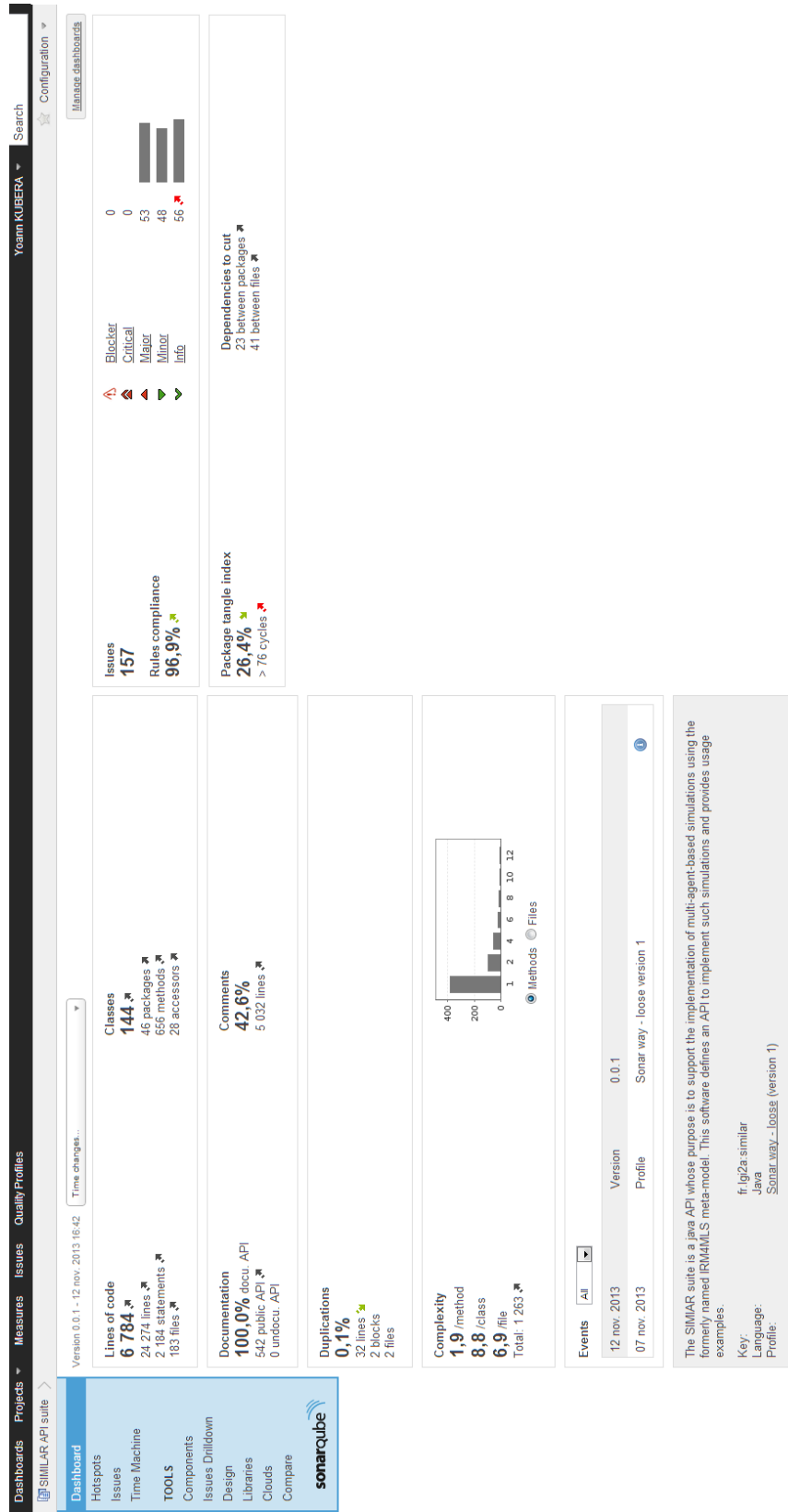


FIGURE 13 – Screenshot de SonarQube : Ecran montrant les différentes métriques

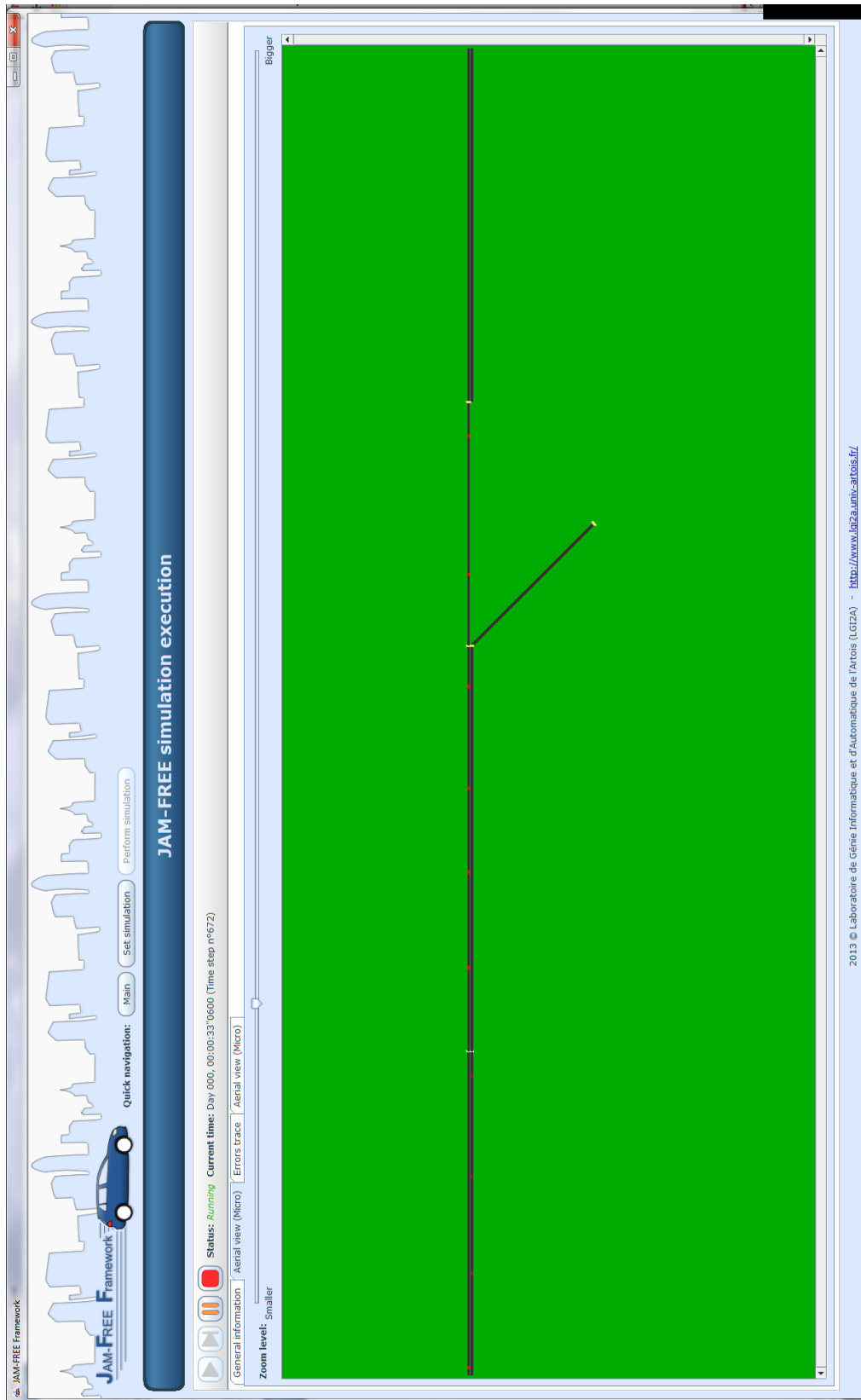


FIGURE 14 – Screenshot de JFF : exécution d'une simulation(debug mode)

## C SonarQube

SonarQube est une plate-forme open source pour l'inspection continue (CI) de la qualité du code. Il calcule de nombreux paramètres de qualité telles que le pourcentage de cas couverts par les tests unitaires. Comme les fig. 12 et 13 le montrent, SIMILAR obtient d'excellents rapports sur SonarQube.

### Références

- BOURREL, E. et LESORT, J. (2003). Mixing micro and macro representations of traffic flow : a hybrid model based on the LWR theory. *82th Annual Meeting of the Transportation Research Board*.
- EL HMAM, M. (2006). *Contribution à la modélisation et à la simulation hybride du flux de trafic*. Thèse de doctorat, Université d'Artois.
- ESPIÉ, S., GATTUSO, D. et GALANTE, F. (2006). Hybrid traffic model coupling macro- and behavioral microsimulation. *In 85th Annual Meeting of Transportation Research Board*, Washington D.C.
- FERBER, J. et MÜLLER, J.-P. (1996). Influences and reaction : a model of situated multiagent systems. *In 2nd International Conference on Multi-agent systems (ICMAS'96)*, pages 72–79.
- GAUD, N., GALLAND, S., GECHTER, F., HILAIRE, V. et KOUKAM, A. (2008). Holonic multilevel simulation of complex systems : Application to real-time pedestrians simulation in virtual urban environment. *Simulation Modelling Practice and Theory*, 16:1659–1676.
- GIL-QUIJANO, J., HUTZLER, G. et LOUAIL, T. (2010). Accroche-toi au niveau, j'enlève l'échelle : Éléments d'analyse des aspects multiniveaux dans la simulation à base d'agents. *Revue d'Intelligence Artificielle*, 24(5):625–648.
- GIL-QUIJANO, J., LOUAIL, T. et HUTZLER, G. (2012). From biological to urban cells : Lessons from three multilevel agent-based models. *In DESAI, N., LIU, A. et WINIKOFF, M., éditeurs : Principles and Practice of Multi-Agent Systems*, volume 7057 de *Lecture Notes in Computer Science*, pages 620–635. Springer.
- KESTING, A., TREIBER, M. et HELBING, D. (2010). Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity. *Philosophical Transactions of the Royal Society A : Mathematical, Physical and Engineering Sciences*, 368(1928):4585–4605.
- MAGNE, L., RABUT, S. et GABARD, J. (2000). Towards an hybrid macro-micro traffic flow simulation model. *Proceedings of the INFORMS Salt Lake City String 2000 Conference*.
- MAMMAR, S. and Lebacque, J. et HAJ-SALEM, H. (2006). Hybrid model based on second-order traffic model. *85th Annual Meeting of Transportation Research Board*, 1(06-2160).
- MICHEL, F. (2007a). The IRM4S model : the influence/reaction principle for multiagent based simulation. *In Proc. of 6th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2007)*, pages 1–3.
- MICHEL, F. (2007b). Le modèle IRM4S. de l'utilisation des notions d'influence et de réaction pour la simulation de systèmes multi-agents. *Revue d'Intelligence Artificielle*, 21:757–779.
- MORVAN, G. (2013). Multi-level agent-based modeling - a literature survey. *CoRR*, abs/1205.0561.
- MORVAN, G. et JOLLY, D. (2012). Multi-level agent-based modeling with the Influence Reaction principle. *CoRR*, abs/1204.0634.



- MORVAN, G. et KUBERA, Y. (2014a). On time and consistency in multi-agent-based simulations. Working paper.
- MORVAN, G. et KUBERA, Y. (2014b). SIMILAR : Simulations with multi-level agents and reactions. Working paper.
- MORVAN, G., VEREMME, A. et DUPONT, D. (2011). IRM<sub>4</sub>MLS : the influence reaction model for multi-level simulation. In BOSSE, T., GELLER, A. et JONKER, C., éditeurs : *Multi-Agent-Based Simulation XI*, volume 6532 de *Lecture Notes in Artificial Intelligence*, pages 16–27. Springer.
- PICAULT, S. et MATHIEU, P. (2011). An interaction-oriented model for multi-scale simulation. In WALSH, T., éditeur : *Twenty-Second International Joint Conference on Artificial Intelligence*, pages 332–337, Barcelona, Spain. AAAI Press.
- POSCHINGER, A., KATES, R. et KELLER, H. (2002). Coupling of concurrent macroscopic and microscopic traffic flow models using hybrid stochastic and deterministic disaggregation. *Transportation and Traffic Theory for the 21st century*.
- SÉTRA (2010). Identification et localisation sur le reseau routier national. Rapport technique, Sétra, 110 rue de Paris 77171 Sourdun - France.
- SEWALL, J., WILKIE, D. et LIN, M. C. (2011). Interactive hybrid simulation of large-scale traffic. *ACM Transaction on Graphics (Proceedings of SIGGRAPH Asia)*, 30(6).
- SOYEZ, J.-B., MORVAN, G., DUPONT, D. et MERZOUKI, R. (2013). A methodology to engineer and validate dynamic multi-level multi-agent based simulations. In *Multi-Agent-Based Simulation XIII*, volume 7838 de *Lecture Notes in Artificial Intelligence*, pages 130–142. Springer.
- VO, A. (2012). *An operational architecture to handle multiple levels of representation in agent-based models*. Thèse de doctorat, Université Paris VI.
- VO, D.-A., DROGOUL, A. et ZUCKER, J.-D. (2012a). An operational meta-model for handling multiple scales in agent-based simulations. In *International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF)*, pages 1–6, Ho Chi Minh City. IEEE.
- VO, D.-A., DROGOUL, A., ZUCKER, J.-D. et HO, T.-V. (2012b). A modelling language to represent and specify emerging structures in agent-based model. In DESAI, N., LIU, A. et WINIKOFF, M., éditeurs : *Principles and Practice of Multi-Agent Systems*, volume 7057 de *Lecture Notes in Computer Science*, pages 212–227. Springer.