

---

# RAFALE-SP : un outil de modélisation pour la simulation de la mobilité

## Vers une modélisation intuitive du déplacement

Nicolas Marilleau, Christophe Lang, Pascal Chatonnay et Laurent Philippe

Laboratoire d'Informatique de l'université de Franche-Comté  
CNRS FRE2661  
16 route de Gray  
25030 Besançon CEDEX  
{nicolas.marilleau,christophe.lang}@lifc.univ-fcomte.fr  
{pascal.chatonnay,laurent.philippe}@lifc.univ-fcomte.fr

---

*RÉSUMÉ.* Nous présentons dans cet article une méthode multi-agent appelée RAFALE-SP orientée mobilité. Elle vise à faciliter la démarche de modélisation pour la simulation des déplacements de mobiles qui évoluent sur un territoire concret tel qu'une ville. Pour cela, elle utilise plusieurs formalismes (UML et Ploom-Unity) ainsi qu'une bibliothèque de simulation. Le tout gravite autour d'une méthode comprenant les étapes nécessaires à la modélisation et à la simulation d'un territoire avec sa complexité et les acteurs de la mobilité.

Dans le cadre de cet article nous nous focalisons plus particulièrement sur un aspect clef de RAFALE-SP : la modélisation du déplacement. Chaque mobile est représenté par un agent pour lequel sont définis des connaissances, des désirs, une perception et une capacité limitée à se mouvoir dans l'environnement. A chaque mobile est attribué un ensemble de vecteurs de déplacement représentatif de la nature des mouvements qu'ils accomplissent dans le monde réel. Ils sont sujet à des contraintes précisément définies afin de représenter au mieux le système étudié.

*ABSTRACT.* The goal of our work is to create a mobility oriented framework called RAFALE-SP. It aims at helping us to design and to simulate mobile motions which take place on a real area like a town. This framework is based on several languages (UML and Ploom-Unity) and a simulation toolkit. These tools are used according to a methodology which contains every step to design and simulate an area according to its complexity and to mobiles.

In this paper, we focus, more particularly, on a key-point of this framework : the design of mobile journey. Each mobile is described by an agent. Each agent have beliefs, desires, a perception and limited capabilities to move on the environment. We associate each mobile with a set of motion vectors which are representative of journey types that studied mobiles can do in the reality. These motion vectors are constrained in order to describe better mobile motion abilities.

*MOTS-CLÉS :* Agent, Déplacement, Méthode, Mobilité, Modélisation, Simulation

2 1<sup>re</sup> soumission à *Journées Francophones sur les Systèmes Multi-Agents 2006*.

*KEYWORDS: Agent, Journey, Methodology, Mobility, Design, Simulation*

---

## 1. Les systèmes multi-agent et la simulation de la mobilité

L'étude des mobilités qui se déroulent sur un territoire réel, notamment celles des hommes, a connu, à la venue des systèmes multi-agents au cours des années 1990, un véritable bond en avant [DAU 04]. Les outils utilisés jusqu'à lors, comme les modèles mathématiques discrets ou continus, ne proposent qu'une vision macroscopique du territoire. La focalisation au niveau de l'individu n'était alors impossible.

Les SMA se sont tout naturellement présentés comme une solution permettant une démarche "*individu centré*" dont l'objectif est de modéliser chaque individu pour simuler et observer les dynamiques globale de la population étudiée. Dans ce contexte, il existe aujourd'hui une multitude de simulateurs dédiés à la simulation des dynamiques des hommes, par exemple Transim [Tra03], d'autres s'appliquent dans divers domaines comme l'écologie comme [LAC 05].

Parallèlement à ces outils, des bibliothèques multi-agent ont vu le jour. Leur objectif est de fournir les briques de base facilitant le développement de logiciels ou de simulateurs. Certaines comme Madkit [GUT 00] sont génériques et peuvent être utilisées dans une grande variété de domaines. D'autres comme Cormas [LEP 00] sont spécifiques au monde de la simulation.

Le constat que nous pouvons faire aujourd'hui est que les SMA sont vus comme une approche de simulation dans le cadre de l'étude des mobilités. Cependant, un certain nombre d'outils de simulation comme Minus [VAN 03] et Padi-simul [MOU 04] sont couplés avec une démarche de modélisation qui s'applique en amont. Ces plateformes proposent à la fois des outils conceptuels et logiciels pour la modélisation et la simulation de territoires. Mais, leurs utilisations restent relativement ciblées pour des domaines particuliers, par exemple, Minus se focalise sur l'étude des dynamiques urbaines.

Nous proposons *une approche de modélisation pour la simulation de la mobilité* qui s'opère sur un environnement réel. Cependant, notre démarche ne s'applique pas uniquement à la modélisation d'un type de territoire particulier : nous souhaitons pouvoir l'utiliser dans un large panel de situation allant de l'aménagement du territoire à la robotique.

La plate-forme de modélisation-simulation que nous présentons dans cet article baptisée RAFALE-SP <sup>1</sup> est suffisamment générique pour modéliser et simuler des mobiles évoluant dans des endroits quelconques comme une ville ou un territoire hostile (ex. une planète extra-terrestre). Pour cela, elle comporte une panoplie de méta-modèles basés sur les formalismes UML et PLOOM-UNITY, et une bibliothèque de simulation construite à partir de MadKit. Le tout gravite autour d'une méthode de modélisation-simulation de la mobilité de "bout en bout" prenant en charge l'utilisateur dès ses premiers pas dans la conception du système jusqu'à la simulation.

---

1. Acronyme pour Réflexion, Analyse, Formalisation d'Agents Localisés dans un Espace pour une Simulation des Pérégrinations

La mobilité est alors explicitée, dans RAFALE-SP, par le biais d'une simulation multi-agent. Chaque mobile est alors préalablement modélisé par un agent évoluant dans un environnement virtuel représentatif d'un territoire avec toute sa complexité notamment avec ses dynamiques internes (ex trafic routier interne à une ville). Les agents sont dotés d'aptitudes limitées de déplacement et de capacités cognitives comme l'apprentissage et la perception partielle et biaisée du monde qui les entoure.

Dans le cadre de cet article, nous nous focalisons sur une partie clefs de RAFALE-SP : la modélisation des capacités de déplacement des mobiles (les autres parties étant présentés dans [MAR 05]). Les mobiles effectuent, sous contraintes, des déplacements dont nous définissons la nature.

Avant de présenter la stratégie de modélisation du déplacement que nous avons adoptée, il est important d'avoir une vision complète de RAFALE-SP. C'est pourquoi, nous nous intéressons à la démarche dans sa globalité dans un premier temps avant de nous focaliser ensuite sur la modélisation du déplacement. Nous étayons nos propos par un cas concrets d'application le projet MIRO [BAN 05].

## **2. RAFALE-SP : Une approche de modélisation-simulation orientée mobilité**

Au cours de cette section, nous nous attachons à décrire les tenants et les aboutissants de RAFALE-SP. Pour cela, nous présentons dans un premier temps les concepts clefs de notre approche. Nous nous focalisons ensuite sur la méthode que nous avons élaborée pour en déduire à quel moment la modélisation du déplacement intervient.

### **2.1. Concepts clefs de RAFALE-SP**

Les entités étudiées sont des mobiles quelconques évoluant dans un espace que nous définissons. Ils ont cependant une connotation humaine car nos travaux s'articulent autour de projets de géomatique s'intéressant aux déplacements des hommes.

Nous modélisons chaque mobile par un agent très humanisé qui suit les caractéristiques des agents VON-BDI (Value, Obligation, Norm-Belief, Desire, Intention) [BEA 02] : entité spécifique créée afin de représenter au mieux la nature humaine. Les mobiles sont alors dotés d'une connaissance évolutive intégrant des concepts de norme et d'habitude. Ces entités autonomes possèdent un certain nombre de désirs à partir desquels sont extraites des intentions de déplacement. Les mobiles disposent aussi d'une perception individuelle et limitée par des contraintes souvent synonymes d'une restriction de la vision à un voisinage proche. A chaque mobile est associé une capacité à se déplacer soumise elle aussi à des contraintes.

Les agents sont localisés dans un espace géographique représentatif d'un territoire réel, par exemple une ville. La mobilité de l'agent se traduit alors par un changement de position dans cet espace. Cette évolution est le résultat d'un calcul complexe qui

s'effectue entre la perception de l'agent, ses connaissances, ses capacités motrices et ses règles internes de comportement.

L'espace est défini selon un référentiel adéquat pour l'étude, autrement dit un découpage du monde nous permettant de positionner les mobiles de façon unique et optimal afin d'observer au mieux la mobilité qui nous intéresse. Ainsi, nous sommes en mesure de modéliser un territoire par un espace géographique structuré sous différentes formes comme un automate cellulaire, un graphe ou encore un monde virtuel à deux ou trois dimensions.

Les territoires que nous modélisons sont fortement contraints de par leur structure (ex. la voirie canalise le déplacement des véhicules) mais aussi à cause des règles qui sont appliquées dessus (ex. le code de la route). En conséquence, les facultés de déplacement des agents sont limitées : un mobile ne peut pas se déplacer de tout point en tout point de l'espace sans passer par des positions intermédiaires. Des aptitudes physiques sont associées à chaque mobile déterminant la nature des déplacements qu'ils peuvent accomplir. Ainsi nous sommes en mesure de modéliser un homme capable de marcher, de courir. Ces aptitudes sont limitantes dans le sens où elles définissent la nature des déplacements. Elles sont aussi limitées car elles ne peuvent pas s'appliquer dans n'importe quelle situation.

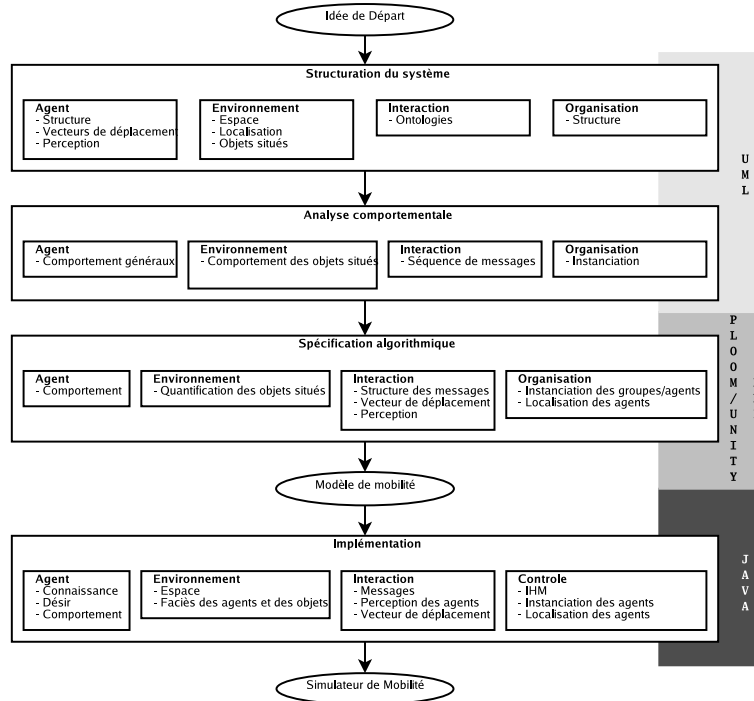
En résumé, nous modélisons la mobilité d'un territoire selon quatre éléments de base l'*Agent*, l'*Environnement*. Nous représentons aussi les *Interactions* entre les mobiles et l'*Organisation* en groupes des mobiles pour la modélisation des entités sociales comme la famille, des entreprises. RAFALE-SP s'inscrit donc complètement dans l'approche VOYELLE (AEIOU) proposée dans [DEM 03]. La notion d'*Utilisateur* est prise en compte dans le toolkit de RAFALE-SP mais aucun formalisme spécifique n'est établi dans le méta-modèle.

L'agent, l'environnement, les interactions et les groupes sont modélisés dans un premier temps et ensuite implémentés dans un simulateur selon une démarche progressive basée sur un méta-modèle et une bibliothèque multi-agent. Cette démarche a pour vocation de guider les pensées du concepteur dans son étude.

## 2.2. RAFALE-SP : une méthode...

La démarche associée à RAFALE-SP propose l'utilisation de manière méthodique de divers formalismes et d'un *toolkit*. Son objectif est de faciliter le travail du scientifique dans l'analyse, la conception et la simulation des mobiles. Il s'agit pour nous d'un guide qui suit les étapes usuelles de création d'une application informatique. Après la rédaction d'un cahier des charges, la méthode se décompose en quatre étapes (voir la figure 1) :

1) *L'analyse structurelle* donne une vision générale de l'architecture de SMA. Cette étape se focalise principalement sur une description structurelle de l'environnement, des mobiles, des groupes et des différents types d'interactions présentes entre les



**Figure 1.** *La méthode.*

agents. Le langage UML est alors utilisé en conséquence de sa popularité et des possibilités qu'il offre. Les diagrammes de classes et de collaborations décrivent, respectivement, la structure des mobiles et les différents types d'interactions qui s'opèrent entre les agents.

2) *L'analyse comportementale* déclare les principales règles de comportements qui seront implémentées dans les agents. Cette étape se focalise aussi sur la définition des protocoles d'interaction entre les différents acteurs (mobiles et utilisateurs). Cette étape nécessite l'utilisation de diagrammes UML comme les diagrammes d'activité et d'état transition. Nous représentons ainsi l'enchaînement des règles de comportements des agents. Le diagramme de séquence définit quant à lui les protocoles d'interaction.

3) *La spécification* est une description fine du système. Pour cela, cette étape encourage la spécification des groupes, la définition de manière algorithmique des différentes règles de comportement des mobiles ainsi que la structuration des messages envoyés entre les différents agents du système. Nous employons alors XML pour la description des messages et un langage formel dédié appelé *Ploom-Unity*. Ce dernier tire partie de deux langages orientés : *Ploom* [FER 90] parce qu'il possède les concepts d'héritage et de fonction, et de *Mobile Unity* [ROM 03] car il renferme des éléments de localisation des agents. A l'issue de la spécification, nous obtenons un modèle de

mobilité pour un système donné. Il ne s'agit pas encore d'un simulateur.

4) *L'implémentation* consiste en la création d'un simulateur dédié qui est la représentation du modèle de mobilité décrit précédemment. Pour cela nous utilisons une bibliothèque de création de simulateur de mobilité qui a été conçue à partir de la bibliothèque multi-agents MadKit [GUT 00]. L'étape de développement n'achève pas le processus de création d'un simulateur. En effet comme tout logiciel, il est nécessaire de le tester mais surtout de le calibrer.

La méthode RAFALE-SP comporte les principaux éléments nécessaires pour la modélisation et ensuite la simulation de mobiles en vue d'une étude des pérégrinations. Cet outil est orienté mobilité ce qui implique une généralité limitée à ce contexte. Cette limite est intentionnelle car notre but n'est pas de faciliter la représentation de tous les SMA.

### 2.3. ... pour la modélisation-simulation des mobilités

La mobilité est, pour nous, constituée d'une succession de déplacements unitaires. Ils se traduisent par des changements de position des mobiles considérés qui sont situés dans l'espace virtuel représentatif du territoire. Ces mouvements s'effectuent en accord avec les aptitudes physiques et intellectuelles attribuées à chaque entité et en fonction des lois qui régissent sur le territoire (ex. code de la route)

RAFALE-SP contient toutes les briques nécessaires pour représenter les déplacements possible de chacun des mobiles, et cela de manière méthodique : le mouvement est modélisé au cours de deux étapes de notre démarche (l'analyse structurelle et la spécification algorithmique) pour être implémenté ensuite. La modélisation complète du déplacement nécessite construction de plusieurs modèles conceptuel mis en relation grâce à une spécification :

- *Un diagramme UML de classe de l'environnement* structurant le territoire ;
- *Un diagramme UML de classe de l'agent* structurant le mobile et ses aptitudes à se mouvoir sur l'environnement ;
- *Une spécification en Ploom-Unity* qui détermine avec précision les interactions qui s'opèrent entre l'agent (représentation du mobile) et son environnement contenant l'espace représentatif du territoire, pour accomplir les différents déplacements.

Les interactions entre les agents représentant les mobiles et leur environnement sont fortes. En effet, chaque déplacement de l'agent donne lieu à des interactions avec son environnement. Ces dernières sont basées autour d'un protocole sûr, déjà établi [ODE 02] qui se déroule en deux temps : l'agent effectue une action sur l'environnement et obtient en retour une nouvelle perception de l'espace d'évolution.

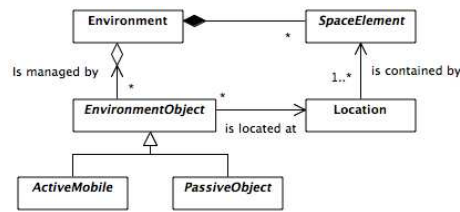
Dans le cadre de notre démarche, les actions sont des intentions de déplacement que nous appelons *Vecteur de déplacement*. Ce nom est analogie avec les mathématiques et les sciences physiques car, nous considérons un vecteur de déplacement comme une transformation de la position d'un mobile en une autre.

Un vecteur de déplacement constitue un ordre de déplacement du mobile dans son espace. Il s'applique sur l'environnement en fonction d'un algorithme défini par la spécification Ploom-Unity qui tient compte de la structure de l'environnement préalablement établie

### 3. Le déplacement dans RAFALE-SP

Trois modèles sont donc nécessaires pour représenter et attribuer des vecteurs de déplacement aux mobiles. Nous les présentons successivement dans cette section. Nous montrons dans un premier temps comment l'environnement est structuré pour nous focaliser ensuite sur la création et la spécification des vecteurs de déplacement.

#### 3.1. Structuration de l'environnement



**Figure 2.** Architecture de l'environnement.

La structuration d'un territoire s'effectue au moyen d'un meta-modèle qui définit les fondations de l'environnement du système multi-agent moteur des simulations futures. Cet environnement doit représenter au mieux le territoire réel avec sa topologie et ses dynamiques internes<sup>2</sup>. Sa modélisation nécessite (voir la figure 2) :

- *la structuration de l'espace* qui revient à définir la topologie du monde où évoluent les mobiles du système. Par exemple, la voirie d'une ville peut être par des objets géographiques organisés selon un graphe orienté.

- *la définition de la position* qui détermine comment les mobiles sont localisés dans l'espace. par exemple, les mobiles peuvent être localisés par des coordonnées cartésiennes à deux ou trois dimensions. D'autres systèmes de références, plus exotique, sont aussi possible.

- *la caractérisation des objets localisés* correspond à l'ensemble des éléments situés dans l'espace qui sont sujet à la perception des mobiles. Nous distinguons deux types d'objets localisés. (i) D'une part, les objets passifs (*PassiveObjects*) permettant, par exemple, la représentation du bâti, de la signalisation. Ces objets ne sont pas autonomes et ont un comportement simple qui se résume une succession d'états modifiés

2. Les dynamiques internes constituent un trafic présent sur le territoire et perturbateur qu'il faut prendre en compte pour une simulation réaliste. Cependant, il ne fait pas l'objet de l'étude



par l'environnement au cours de la simulation (ex. un restaurant peut être ouvert ou fermé selon l'heure de la journée). (ii) D'autre part, les objets actifs ou *ActiveObjects* correspondent aux mobiles simulés. Chaque *ActiveObject* est associé à un agent qui a la charge de le faire évoluer.

Les *ActiveObjects* peut être vus comme un corps, un faciès, des mobiles dans l'espace. Pour se déplacer, les agents associés à chacun d'eux génèrent des vecteurs de déplacement conduisant au changement de la position du faciès du mobile. Le vecteur de déplacement est donc dans RAFALE-SP le sésame du mouvement.

### 3.2. Définition des vecteurs de déplacement

Un vecteur déplacement est une action que l'agent est susceptible d'accomplir pour se déplacer sur l'environnement virtuel.

Plusieurs vecteurs de déplacement peuvent être associés à un même mobile. Ils est ainsi capable de faire des déplacements de natures différentes (ex. marcher ou courir) en fonction de sa situation sur le territoire et de son état mental.

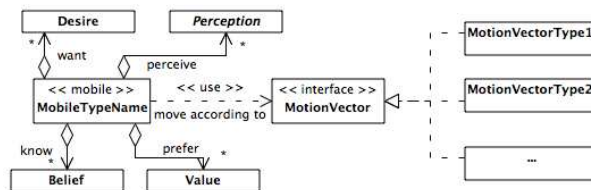


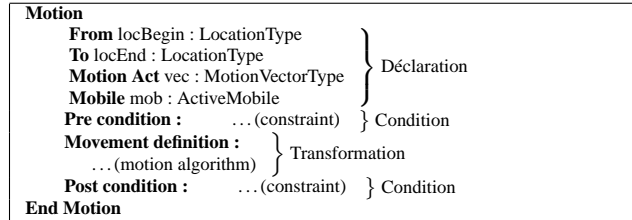
Figure 3. Extrait d'un diagramme structurel du mobile.

Les vecteurs de déplacement sont définis dans le diagramme de classe UML décrivant la structure du mobile (voir la figure 3). Chacun d'eux est représentés par une classe qui implémente l'interface *MotionVector*. Le nom de la classe caractérise la nature du déplacement (ex sauter), les attributs traduisent la porté du déplacement dans l'espace, la taille du mouvement (ex. longueur du saut en mètre).

En résumé, d'un côté, un diagramme UML traduit la structure de l'environnement et de quelle manière les agents sont localisés dessus. De l'autre, différents diagrammes UML présentent la structure des mobiles (ex connaissances, buts) mais aussi un certain nombre de *MotionVector*. Jusqu'à lors, nous précisons, à aucun moment, comment un vecteur de déplacement modifie la position d'un mobile lorsqu'il est appliqué par ce dernier.

### 3.3. Spécification des vecteurs de déplacement

La spécification du déplacement joue un rôle double : (i) elle décrit, d'une part, grâce à un algorithme, comment la position d'un mobile est modifiée en fonction



**Figure 4.** Structure d'une spécification d'un vecteur de déplacement

d'un vecteur de déplacement généré par l'agent ; (ii) elle vérifie d'autre part que le déplacement du mobile n'entraîne pas un paradoxe dans la simulation, par exemple, elle s'assure qu'un mobile ne soit pas localisé hors des frontières du territoire simulé.

Une spécification est exprimée selon le langage Ploom-Unity : langages spécialisé dans la description fine d'agents localisés et mouvants dans un espace géographique. Ce langage suit une grammaire que nous avons définie selon un format EBNF. Nous l'avons testée avec JAVACC, notamment la partie qui concerne la spécification des déplacements, afin de lever toutes les ambiguïtés dans le langage.

Le langage Ploom-Unity décompose en trois parties la description des vecteurs de déplacement :

- *L'entête* déclare quatre variables dont la position avant le déplacement (*locBegin*), la position après le déplacement (*locEnd*), le vecteur de déplacement origine du mouvement (*vec*) et le mobile déplacé (*mob*). Ces variables sont utilisés par la suite dans la spécification au sein de l'algorithme de déplacement.

- *L'algorithme* déduit, par le calcul, la position d'arrivée du mobile à partir de sa position de départ et du vecteur déplacement associé. Autrement dit, son rôle est d'initialiser la variable *locEnd*, représentative de la nouvelle position du mobile après le mouvement.

- *Les conditions* sont des points de validation qui assurent l'intégrité du mouvement sur le territoire avant (grâce à la pré condition) et après (grâce à la post condition) son application.

Les pré et post conditions sont employées dans une spécification pour des fins différentes. L'une, la pré condition, vérifie que le vecteur de déplacement est applicable dans la situation courante de l'agent. La post condition est un moyen de s'assurer qu'après déplacement, le mobile ne se trouve pas dans une situation incohérente.

La spécification du déplacement est donc un lien algorithmique entre les vecteurs de déplacement générés pas les mobiles et sa position dans son espace d'évolution. Elle décrit très fidèlement comment un vecteur de déplacement modifie la position de l'agent qui l'utilise pour se mouvoir.

#### 4. Cas d'étude : le projet MIRO

Nos travaux de recherche s'organisent principalement autour d'un projet appelé MIRO [BAN 05]. Il s'agit d'un projet pluridisciplinaire associant des géographes et des informaticiens dans un but commun : l'étude de "la fourmilière urbaine". Cette approche centrée sur l'individu s'oriente vers une description de la personne pour une étude de la population et une observation de ses dynamiques en milieu urbain. Le lieu d'étude choisi est la ville de Dijon.

Les spécificités des mobiles étudiés sont extraites à partir d'une enquête de terrain. Cette enquête contient des données statistiques sur la population de Dijon. Nous les désagrégeons afin de générer une population synthétique représentative de la réalité. Cette dernière sera ensuite simulée pour, dans un premier temps, valider notre étude et ensuite émettre des hypothèses. En d'autres termes, nous voulons vérifier que les mobiles que nous créons, reproduisent correctement la réalité. Ensuite nous désirons modifier leur milieu d'évolution (la ville) pour étudier leurs dynamiques.

A partir des données de l'enquête, un certain nombre de classes de mobiles sont extraites. Elles définissent de manière statistique les tâches que les individus ont à accomplir dans la journée. La population synthétique est alors générée en respectant des proportions réelles. Pour chaque instance de mobile, un certain nombre de tâches lui sont affectées. Il est alors en mesure de calculer un programme d'activité avant de se déplacer sur le territoire pour accomplir ses tâches.

##### 4.1. La ville virtuelle

Nous caractérisons la ville de Dijon, territoire où évoluent les mobiles, par sa voirie ainsi qu'un ensemble de services proposés aux mobiles (lieux de travail, loisir, magasin, etc) que nous insérons dans le bâti. Les mobiles se déplacent dans les rues de la ville virtuelle et sont capable de percevoir les services qui leurs sont offerts.

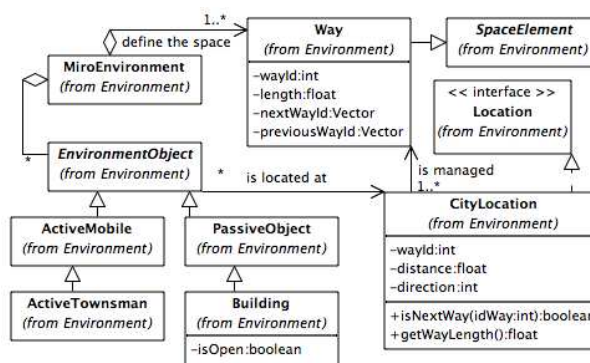


Figure 5. Structure de l'environnement dans le projet MIRO

La première étape dans la modélisation de la ville vise à structurer le territoire. Pour cela, le réseau routier est traduit sous la forme d'un graphe orienté où chaque arc représente un tronçon paramétré par un ensemble de constante (son identifiant, sa longueur et le nombre de voies pour chacun des sens de circulation, le noeud de départ et le noeud d'arrivée).

Les mobiles et le bâti sont localisés dans l'espace selon un numéro de tronçon, une direction (l'agent va du noeud de départ au noeud d'arrivée ou *vis et versa*) et la distance qui sépare le mobile du noeud de départ. Ces informations sont synthétisées au sein d'une même classe représentant la position que nous avons appelée *CityLocation*

Le bâti est décrit par une classe appelée *Building*. Il est caractérisé par un ensemble de services accessible par les mobiles à certaines heures de la journée. L'état de ces lieux d'activités (ouvert, fermé) est modifié par l'environnement. L'accessibilité du service est perçue par le mobile qui s'adapte en fonction pour accomplir ses tâches.

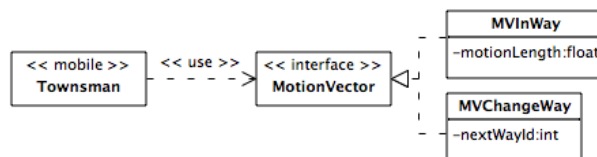
Chaque mobile est quant à lui représenté dans l'espace par un objet de type *ActiveTownsmen*. Cet élément est assimilable au faciès de l'agent : il détermine ce que les autres peuvent percevoir de lui. Il est donc associé à un agent dont l'objectif est de reproduire le comportement du citoyen.

#### 4.2. Les citoyens et leurs capacités de déplacement

Les agents représentatifs des citoyens sont animés par des règles de comportement préalablement définies. Ces dernières se bases sur les connaissances de l'agent, ses objectifs et ses perceptions, pour déplacer le mobile selon des aptitudes physiques.

Les mobiles se déplacent de manière très rudimentaires. Seulement deux vecteurs de déplacement sont nécessaire pour évoluer dans tout l'espace (figure 6) :

- *MVInWay* pour effectuer un déplacement au sein d'un même tronçon. Ce dernier prend en paramètre la longueur en mètre du déplacement.
- *MVChangeWay* pour que le mobile puisse changer de tronçon. Ce vecteur de déplacement possède comme unique donnée membre l'identifiant du tronçon suivant.



**Figure 6.** Extrait du modèle structurel des mobiles.

L'emploi d'un ce des deux vecteurs de déplacement par un agent entraîne changement de position dans l'espace selon un algorithme sous certaines contraintes que nous définissons au sein d'une spécification.

Par exemple, le vecteur *MVInWay* conduit simplement à modifier le champ *distance* de la position du mobile en fonction de la direction du déplacement (voir la figure 7). Si le mobile se déplace du début du tronçon à la fin, alors la longueur du déplacement est ajoutée à la distance séparant le mobile du noeud de départ. Sinon, la longueur est soustraite. L'utilisation du vecteur de déplacement *MVInWay* est limitée par deux conditions. La pré condition assure que le mobile est correctement localisé sur un tronçon. La post condition vérifie que la position du mobile après déplacement soit toujours valide.

```

Motion
  From locBegin : CityLocation
  To locEnd : CityLocation
  Motion Act vec : MVInWay
  Mobile mob : Townsman
  Pre condition : locBegin.distance ≥ 0 ∧ locBegin.distance ≤ locBegin.getWayLength()
  Movement definition :
    locEnd.wayId=locBegin.wayId
    locEnd.direction=locBegin.direction
    //The mobile moves from the begin to the end of a way
    if(direction==0){ locEnd.distance=locBegin.distance+vec.length }
    //The mobile moves from the end to the begin of a way
    else { locEnd.distance=locBegin.distance-vec.length }
  Post condition : locEnd.distance ≥ 0 ∧ locBegin.distance ≤ locEnd.getWayLength()
End Motion

```

**Figure 7.** Spécification du vecteur de déplacement *MoveInWay*.

La spécification du déplacement clôture la phase de modélisation. Nous obtenons, ce que nous appelons un *modèle de mobilité* qui n'est autre qu'une représentation agent d'un territoire géographique avec tous ses acteurs. Il s'agit d'un modèle qualitatif du système qui nécessite une simulation pour obtenir des résultats. Cela implique l'utilisation d'un simulateur adapté.

### 4.3. Implémentation du déplacement

La création d'un simulateur est facilitée par l'utilisation d'une bibliothèque JAVA que nous avons développée et intégrée dans RAFALE-SP. Il s'agit d'un embryon de simulateur de mobilité. Elle doit être surchargée en fonction du modèle de mobilité établi lors des étapes précédentes.

Par exemple, l'implémentation du vecteur de déplacement *MVInWay* présenté dans la figure 6 et spécifié ensuite dans la figure 7, nécessite la création d'une classe JAVA qui porte le même nom, et qui renferme les mêmes attributs. Cette classe doit implémenter trois méthodes : l'une contenant l'algorithme de déplacement ; les deux autres renferment les pré et post conditions.

La programmation d'un simulateur suit donc complètement le philosophie du méta-modèle ce qui rend aisée la transformation d'un modèle de mobilité en un simulateur. Il est à noter que la génération automatique d'un simulateur à partir d'un modèle est possible. Cependant, la création d'un tel outil nécessite un important effort d'ingénierie.

Les simulateurs générés utilisent la plate-forme multi-agents MadKit [GUT 00] comme moteur pour la gestion des cycles de vie des agents et des communications inter agents. L'exécution d'une simulation peut être répartie sur une grappe d'ordinateurs. Chacune des machines se voit alors affecter la charge d'une partie du territoire et d'une partie des mobiles simulés. De plus, des stratégies de répartition de charge sont possibles car les agents représentant les mobiles ont la faculté de pouvoir se déplacer de serveur en serveur (mobilité de code simplifiée).

Cette bibliothèque est testée dans le cadre du projet MIRO. En effet, un simulateur est en cours de développement. Il permet déjà de montrer l'évolution de plusieurs milliers de mobiles qui se déplacent dans une ville virtuelle constituée à partir de données réelles provenant d'une base de données géographique MySQL.

## 5. Conclusion

Outil de modélisation pour la simulation de la mobilité, RAFALE-SP présente tous les atouts nécessaires pour décrire le mobile dans son individualité (ex piéton, automobiliste) et son territoire d'évolution avec toute sa complexité (ex la ville).

Chaque mobile est modélisé par un agent cognitif qui se voit affecter un ensemble de *vecteurs de déplacement* représentatifs de la nature des mouvements qu'il est en mesure d'accomplir dans la réalité (marche, conduite). Ces vecteurs de déplacement sont soumis à des contraintes à définir au cas par cas, pour limiter leur portée sur l'environnement virtuelle et ainsi représenter au mieux un mouvement réel.

RAFALE-SP propose donc une approche intuitive de modélisation du déplacement en analogie avec la réalité, qui s'effectue au travers de divers méta-modèles orientés mobilité. Ces derniers sont utilisés méthodiquement afin de décrire les mobiles par des agents et le territoire comme un environnement virtuel complexe disposant de sa propre dynamique interne, et contraint. Une bibliothèque de simulation est aussi employée lors de la dernière phase de notre démarche afin de faciliter le développement de simulateurs sources de résultats concrets.

La démarche que nous proposons peut être qualifiée de bout en bout car son application débute dès les prémisses de la modélisation des mobiles et du territoire, et se termine par leur simulation. Dans le cas présent, une intervention humaine est nécessaire entre chaque étape de notre méthode afin de transcrire un modèle en un autre. Cependant, une transcription semi automatique d'un modèle en un autre nous semble possible.

Compte tenu de l'état d'avancement de nos travaux, de nouvelles perspectives s'ouvrent à nous : la création d'un *laboratoire artificiel de la mobilité* disposant de tous les outils logiciels nécessaires à la conception de modèles de mobilité, à la génération semi automatique voire automatique de simulateurs et à la vérification des modèles et simulateurs générés par un système de *model checking*. Ce type d'outils présente un intérêt réel dans le cadre de la robotique et des micro-techniques.

## 6. Bibliographie

- [BAN 05] BANOS A., CHARDONNEL S., LANG C., MARILLEAU N., THEVENIN T., « Simulating the swarming city : a MAS approach », *The 9th Int. Conf. on Computers in Urban Planning and Urban Management, CUPUM 2005*, London, UK, 2005.
- [BEA 02] BEAVERS G., HEXMOOR H., « In Search Of Simple And Responsible Agents », *The GSFC Workshop On Radical Agents*, McLean, VA, USA, 2002, p. 257-268.
- [DAU 04] DAUDÉ E., « Apport de la simulation multi-agents à l'étude des processus de diffusion », *Cybergeo : Revue européenne de la géographie*, vol. 255, 2004.
- [DEM 03] DEMAZEAU Y., « Créativité Emergente Centrée Utilisateur », *11èmes Journées Francophones sur les Systèmes Multi-Agents*, Hammamet, 2003, Hermès, p. 31-36.
- [FER 90] FERBER J., *Conception et programmation par objets*, Hermes, 1990.
- [GUT 00] GUTKNECHT O., FERBER J., MICHEL F., « MadKit : une architecture de plateforme multi-agents générique », *Rapport de recherche n° 00061*, 2000, Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier, France.
- [LAC 05] LACOMBE G., BOUSQUET F., KUPER M., BOONMUN C., PHUPAK S., NAIVINIT W., CHIRAWAT V., TRÉBUIL G., « How farmers can manage variability in the rice ecosystem of northeast Thailand ? The contribution of agent based modeling », *International Conference on Simulation and Modelling*, Rose Garden, Bangkok, 2005, p. 345-352.
- [LEP 00] LE PAGE C., BOUSQUET F., BAKAM I., PROTON H., « Cormas : A multiagent simulation toolkit to model natural and social dynamics at multiple scales », *the ecology of Scales*, vol. 1416, 2000, p. 826-838.
- [MAR 05] MARILLEAU N., LANG C., CHATONNAY P., PHILIPPE L., « Cognitive Perception in RAFALES-SP Methodology », *International Conference on Computational Intelligence for Modelling Control and Automation*, Vienna, Autriche, 2005, p. 594-600.
- [MOU 04] MOULIN B., CHAKER W., GANCET J., « PADI-Simul : an agent-based geosimulation software supporting the design of geographic spaces », *Computers, Environment and Urban Systems*, vol. 28, n° 4, 2004, p. 387-420.
- [ODE 02] ODELL J., PARUNAK H. V. D., FLEISCHER M., BREUCKNER S., « Modeling Agents and their Environment », *Agent-Oriented Software Engineering (AOSE) III*, vol. 2585, Berlin, Allemagne, 2002, Lecture Notes on Computer Science, p. 16-31.
- [ROM 03] ROMAN G.-C., PAYTON J., « Mobile UNITY Schemas for Agent Coordination », *ASM 2003*, vol. 2589, Taormina, Italy, 2003, Springer, p. 126-150.
- [Tra03] « Transims, Transportation Analysis Traffic and Simulation Systems », Los Alamos National Laboratory, 2003, Voir [transims.tsasa.lanl.gov](http://transims.tsasa.lanl.gov).
- [VAN 03] VANBERGUE D., « Conception de simulation multi-agents : Application à la simulation des migrations intra-urbaines de la ville de Bogota », *PhD thesis*, LIP6, Paris, 2003.