



**REALISATION D'UN GUIDE D'UTILISATEUR
DU MODELE INTEGRE D'INTERACTION
TRANSPORT ET URBANISME URBANSIM
DANS LE CADRE DU PROJET SIMULER LES
MOBILITES POUR UNE AGGLOMERATION
DURABLE (SIMBAD)**

Auteur : Benjamin SAUBION

VA : Transport et territoire

Promotion : 52

Date : 1^{er} août 2007

Président du jury : M. P. SARDIN
Maître de TFE : M. J-P. NICOLAS
Expert : Mme A. CAMBIEN

NOTICE ANALYTIQUE

	NOM	PRENOM	
AUTEUR	SAUBION	Benjamin	
TITRE DU TFE	Réalisation d'un guide d'utilisateur du modèle intégré d'interaction transport et urbanisme UrbanSim dans le cadre du projet simuler les mobilités pour une agglomération durable (SIMBAD)		
	ORGANISME D'AFFILIATION ET LOCALISATION	NOM PRENOM	
MAITRE DE TFE	Laboratoire d'Economie des Transports	NICOLAS Jean-Pierre	
COLLATION	59 pages	2 Annexes	10
MOTS CLES	Modélisation, modèle, URBANSIM, interaction, transport, urbanisme, localisation, accessibilité, aire urbaine de Lyon, SIMBAD		
TERMES GEOGRAPHIQUES			
RESUME			
ABSTRACT			

SOMMAIRE

1 INTRODUCTION	9
2 PRESENTATION DE LA PLATEFORME URBANSIM, UNE PLATEFORME S'INSERANT DANS UN PROJET GLOBAL : SIMBAD	11
2.1 Un projet s'appuyant sur les expériences passées.	11
2.1.1 Télescopage, à l'origine de SIMBAD.	11
2.1.2 Les évolutions apportées par SIMBAD	12
2.2 L'architecture de SIMBAD, ses différents modules, une organisation basée sur Urbansim.	14
2.3 Le modèle Urbansim	15
2.3.1 Introduction	15
2.3.2 Présentation de l'architecture	16
2.3.2.1 Le module de transition	18
2.3.2.2 Le module de mobilité	18
2.3.2.3 Le module d'accessibilité	19
2.3.2.4 Le module de localisation	20
2.3.2.5 Le module de développement immobilier	21
2.3.2.6 Le module de détermination du prix du foncier	21
3 ANALYSE TECHNIQUE, GUIDE D'UTILISATEUR DU MODELE URBANSIM.23	
3.1 Le travail sur les données	23
3.1.1 Les données d'entrée	23
3.1.2 L'affichage des tables	25
3.1.3 Les données de sortie	27
3.1.4 Les représentations des données	28
3.2 Lancer une simulation	30
3.2.1 Le codage utilisé	30
3.2.2 Etape préalable à la simulation, l'estimation des modèles	31
3.2.3 La simulation	36
3.2.4 Modifier ou créer un modèle	41
3.2.4.1 Au niveau de l'estimation	41
3.2.4.2 Au niveau de la simulation	42
3.3 Conclusion partielle	43

4 CONCLUSION	44
5 BIBLIOGRAPHIE	45

LISTE DES ILLUSTRATIONS

<i>Figure 2-1 : aire d'étude du modèle Télescope (NICOLAS et alii, Rapport n°4 : l'architecture du modèle au sein du projet SIMBAD, Mars 2006)</i>	12
<i>Figure 2-2 : La structure du modèle Urbansim (source :The UrbanSim Project, Juillet 2007)</i>	17
<i>Tableau 1: Liste des fichiers de données d'entrée.</i>	25
<i>Tableau 2 : Caractéristiques d'une cellule de la table "gridcells".</i>	26
<i>Figure 3-1:Illustration de la visualisation d'un extrait de la table "gridcells" après extraction ver le logiciel MySQL.</i>	27
<i>Tableau 3: Liste des fichiers de données de sortie</i>	28
<i>Figure 3-2 : Représentation des cellules à moins de 500m d'une artère routière.</i>	29
<i>Figure 3-3 : Représentation des cellules distantes de moins de 500m d'une artère routière et dont la valeur non résidentielle est inférieure à 22270\$.</i>	30
<i>Tableau 5 : illustration depuis MySQL de la prise en compte des modifications dans le choix des variables spécifiées depuis NLPM_specification.py.</i>	42

1 INTRODUCTION

Au cours de ces trois dernières décennies, la modélisation du transport urbain a progressivement pris en compte les différentes théories et méthodes aboutissant à une structure cohérente et reflétant de façon plus approfondie la situation réelle. Différents types de modèles dynamiques rendant compte des interactions transport et urbanisme se sont alors développés (Hunt *et al.*, 2005).

Ce développement s'est accéléré à la fin des années 90, on trouve des travaux accomplis aussi bien au niveau européen qu'américain. Ainsi, par exemple, 1999 voit le lancement d'UrbanSim aux Etats-Unis tandis qu'au même moment le CERTU développe le projet TRANUS sur la ville Lyonnaise. Cependant, bien qu'ayant apporté certaines avancées, ce projet n'a pas donné entière satisfaction. A l'heure actuelle, il existe peu de résultats en terme de modélisation des interactions transport et urbanisme au niveau français.

L'élaboration de modèles interactifs de transport et d'occupation des sols est devenue une nécessité afin de répondre aux besoins d'évaluation des politiques de transport dans un horizon à long terme. En effet, l'insuffisance de l'approche traditionnelle de la modélisation de la demande de déplacement conduisant à ignorer l'interaction entre transport et occupation des sols a été mise en avant par de nombreux travaux scientifiques (Torrens, 2000 ; Waddell *et al.*, 2001).

De nouveaux modèles sont donc apparus à l'image, au niveau français, des projets SIMAURIF (modèle dynamique de SIMulAtion de l'interaction Urbanisation-transports en Région Ile-de-France) depuis 2003 et SIMBAD (SIMuler les MoBilités pour une Agglomération Durable sur l'air urbaine Lyonnaise) depuis 2005 dans lequel s'inscrit ce Travail de Fin d'Etudes (TFE). Ces modèles ont comme objectif de répondre à cette double problématique régissant les transports urbains : d'une part comment l'urbanisation transforme la mobilité ? et d'autre part, comment la mobilité transforme l'urbanisation ? (Masson, 2000).

Comme nous allons le voir, le projet SIMBAD est basé sur l'architecture du modèle d'urbanisme UrbanSim développé à l'université de Washington. Cependant, ce modèle a été développé sur un modèle urbain américain qui ne répond pas aux mêmes problématiques qu'à une échelle de ville française. Ainsi, un travail sur l'architecture globale du modèle ainsi que sur le recueil et la prise en compte des données doit être fait en amont.

Le présent Travail de Fin d'Etudes (TFE) s'inscrit donc dans ce projet collectif mis en place au Laboratoire d'Economie des Transports commun à l'Ecole Nationale des Travaux Publics de l'Etat et à l'Institut de Sciences Humaines de l'Université Lyon 2. L'objectif de ce TFE est donc d'élaborer un guide d'utilisateur de la plateforme de modélisation UrbanSim afin de relever les éléments importants à prendre en compte. Effectué en amont de l'intégration des données lyonnaises au modèle, ce travail sur les données américaines fournies librement sur le site www.urbansim.org ainsi que sur l'organisation globale du modèle, permettra de mettre en avant les informations à considérer en matière de modélisation dans la perspective de cette future intégration.

Enfin étant donné que le projet SIMBAD se déroule sur plusieurs années, ce TFE, sous forme de guide d'utilisateur, s'inscrit dans le cadre d'un projet scientifique collectif. Il permettra aux personnes s'ajoutant dans le projet d'établir un premier contact avec la plateforme UrbanSim et les problématiques qui l'accompagnent.

2 PRESENTATION DE LA PLATEFORME URBANSIM, UNE PLATEFORME S'INSERANT DANS UN PROJET GLOBAL : SIMBAD

Le travail entrepris au début de ce Travail de Fin d'Etudes (TFE) autour des problématiques d'interactions de transport et urbanisme et de leurs modélisations à l'aide du modèle URBANSIM s'inscrit dans un projet global : SIMBAD, Simuler les MoBilités pour une Agglomération Durable. Ce projet, mis en place par le Laboratoire d'Economie des Transports (LET) en 2005, a pour objectif de mettre en place un modèle prospectif qui rende compte des enjeux environnementaux, économiques et sociaux des déplacements de personnes et de marchandises sur l'aire urbaine de Lyon à l'horizon 2025.

2.1 Un projet s'appuyant sur les expériences passées.

2.1.1 Télescopage, à l'origine de SIMBAD.

SIMBAD s'appuie sur les travaux et les expériences tirés du projet Télescopage. Cependant, la problématique a été réétudiée par rapport à celle développée dans Télescopage, centrée sur les problèmes de congestion engendrés par la superposition de flux de différentes natures sur le noeud lyonnais. Une attention plus particulière est donnée à la prise en compte des interactions transport et urbanisme comme nous allons le voir dans cet exposé.

A l'origine Télescopage est un modèle de simulation à 20 ans basé sur les interactions des différents trafics de marchandises/de personnes, urbain/interurbain. Calibré sur 1995, il donne une image de ces différents trafics sur une aire circulaire de 45 kilomètres autour de Lyon comprenant son aire urbaine (voir Figure 2-1). Télescopage avait une réflexion basé sur l'étude des trafics et de leurs interactions sur ce domaine d'étude. Ses principaux objectifs étaient : « (1) identifier la part respective de chaque segment de demande dans la formation des flux, selon la nature (marchandises, personnes), le mode (route, fer transport urbain collectif) et l'échelle spatiale (locale, interrégionale) de transport ; (2) simuler les effets des déterminants sociaux économiques et des politiques de transport et d'aménagement sur l'augmentation de chaque type de flux ; (3) mettre en exergue les dysfonctionnements du système de transport au niveau d'une « région-ville » ; et (4) acquérir une meilleure compréhension des interactions entre les trafics locaux et les trafics interurbains »¹.

¹ Source : NICOLAS et alii, Rapport n°4 : l'architecture du modèle au sein du projet SIMBAD, Mars 2006

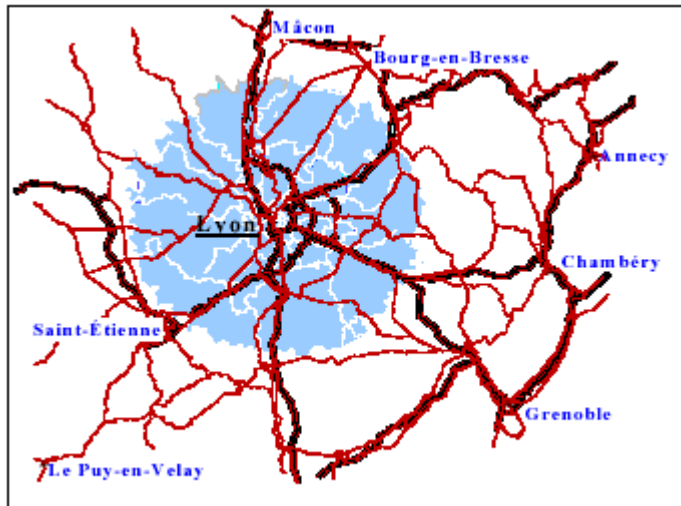


Figure 2-1 : aire d'étude du modèle Telescopage (NICOLAS et alii, Rapport n°4 : l'architecture du modèle au sein du projet SIMBAD, Mars 2006)

Une des principales caractéristiques remises en cause dans ce modèle est l'absence de boucle retour de l'interaction transport à urbanisme dans une vision à long terme. Difficulté mise en avant par la diversité des différents sous-modèles utilisés. En effet, d'une part les quatre sous-modèles présents dans Telescopage n'étaient pas conçus à la base pour s'articuler ensemble autour d'une structure commune ce qui posait problème d'un point de vue méthodologique. D'autre part, les objets modélisés (trafics de personnes/de marchandises, urbain/interurbain, affectation des différents trafics) ne permettaient pas de prendre en compte les effets rétroactifs des transports sur l'urbanisme. Ceci a conduit à de nouvelles considérations : l'occupation des sols influence le système de transports, en retour, la mise en place d'infrastructures de transport joue également un rôle sur la localisation des ménages, des activités, des entreprises et d'une manière générale sur la planification de l'espace urbain. C'est ce lien transport à urbanisme dans une projection à long terme que SIMBAD cherche à intégrer et à étudier, avec comme outils principaux :

- Un modèle de transport en cours de développement sur une base de système à quatre étapes permettant de produire des matrices de coûts généralisés nécessaires à la construction des fonctions d'accessibilité par zone.
- Deux modèles de localisation, résidentielle et des activités économiques reposant sur la structure du modèle d'urbanisme, Urbansim dont la plateforme fera l'objet d'une étude approfondie.

Ainsi, dans la réflexion préalable à SIMBAD, il a fallu s'assurer de la prise en compte de ces problèmes soulevés à l'issue du travail entrepris avec Telescopage. Ces manques relevés ont été considérés et différentes modifications ont été alors apportées.

2.1.2 Les évolutions apportées par SIMBAD

Tout d'abord, à la base de la mise en place du projet SIMBAD, la prise en compte des effets rétroactifs des transports sur l'urbanisme dans une perspective de long terme aura été le premier point abordé et traité suite aux travaux de Telescopage. Seuls les liens urbanisme à transport étaient considérés dans la phase de génération des déplacements du modèle de transport. Or, l'influence en

retour d'une modification de la structure des transports, par la mise en place d'une nouvelle infrastructure ou nouvelle tarification par exemple, peut avoir une importance sur la structure urbaine de l'aire étudiée dans une vision à long terme. Un effort a donc été porté sur la réflexion des localisations des personnes (ménages), et des activités économiques avec la construction de deux modules s'insérant dans le projet global. La notion d'accessibilité liée à l'offre de transport suivant les zones de l'aire d'étude considérées apparaît donc comme le lien permettant ce retour de transport à urbanisme.

Cette optique de vision à long terme des interactions transport et urbanisme a également influencé la structure et l'architecture globale sur laquelle s'organise le projet SIMBAD, moins tourné vers les problématiques de congestion. Les trafics d'échanges et de transit, abordés par les quatre modules de Télescopage, sont considérés dans une moindre mesure. Un véritable travail sur l'intégration des différents sous modules dans le projet global SIMBAD a également été fait. Le choix de la structure sur laquelle le projet reposera a été primordial. Il était indispensable d'éviter les problèmes de connexion et de mise en relation des différents modèles et modules de Télescopage. Le travail avec différents modèles, pas nécessairement créés pour être mis en place ensemble, s'est avéré fastidieux et peu efficace. L'équipe SIMBAD a rapidement eu la volonté de travailler en amont sur l'interface choisie ainsi que sur la gestion des connexions entre les modèles. Il est ressorti de ce premier bilan le choix d'un modèle intégré pour éviter les écueils connus avec Télescopage. Le choix du modèle d'urbanisme Urbansim s'est fait dans ce sens, de par sa flexibilité et des possibilités de prise en compte des différents modèles déjà développés au LET qu'il offrait (DEYMIER, NICOLAS, Juillet 2005).

La thématique de recherche « Modélisation des interactions transport et urbanisme » abordée ici s'inscrit donc dans l'opération de recherche « SIMuler les MoBilités pour une Agglomération Durable (SIMBAD) », menée par le LET. Comme nous l'avons souligné précédemment, elle a pour objectif de mettre en place un modèle prospectif qui rende compte des enjeux environnementaux, économiques et sociaux des déplacements de personnes et de marchandises sur l'aire urbaine de Lyon à l'horizon 2025. SIMBAD a pour finalité de permettre la simulation de différents scénarios de politique des transports et d'urbanisme sur ce territoire et de tester leurs impacts dans ces trois dimensions. En matière de modélisation, cette volonté affichée d'intégrer cette problématique de mobilité durable a amené plusieurs considérations (NICOLAS, Mars 2006) :

- afin de bien prendre en compte et de tester les effets environnementaux, économiques et sociaux des différentes politiques d'urbanisme et de transport mises en place, il était nécessaire d'envisager plusieurs scénarios contextuels. Ceci a pour conséquence directe d'avoir à disposition *un outil permettant de nombreuses modélisations et simulations* visant à révéler les nuances des effets provoqués par tel ou tel scénario. En effet, avec le choix des scénarios d'évolution urbaine et de politiques de régulation alternatives simulés (suivant par exemple les hypothèses d'évolution économique, de mise en place de tarification dans les transports), ce ne seront pas nécessairement les mêmes dimensions qui constitueront l'enjeu.
- Autre point fortement souligné lors de la réflexion préalable au projet autour de la notion de mobilité durable est la volonté de révéler plus précisément *qui réalise les déplacements sur l'aire étudiée*. Dans les modèles comme Télescopage, dans lesquels cette notion n'était pas abordée, les motifs étaient souvent à la base de la génération des déplacements. Dans SIMBAD, il apparaît indispensable d'avoir plus de précisions sur les déplacements et surtout, sur les individus effectuant ces déplacements. Les données socioéconomiques des individus deviennent alors une caractéristique à considérer pour savoir qui est concerné par les différentes mesures mises en place sur l'aire. Ainsi, un travail particulier a été mené par François Gitton sur le module de génération des déplacements avec une attention portée aux boucles de déplacement

(ensemble de déplacements réalisés entre le départ et le retour à domicile) et leur lien qu'elles ont avec les personnes qui en sont à l'origine.

- Enfin, il est important de souligner que le projet SIMBAD s'applique à une *échelle macroscopique* et non microscopique. Il n'est en aucun cas un modèle visant à analyser la mise en place d'une infrastructure particulière sur une zone donnée de l'aire urbaine mais plutôt l'impact que peut avoir une politique au niveau de l'agglomération.

2.2 L'architecture de SIMBAD, ses différents modules, une organisation basée sur Urbansim.

Suite à ces différentes évolutions tirées des expériences du projet Télescope ainsi qu'à la problématique prise en compte, le projet SIMBAD s'organise autour de 6 blocs principaux reliés entre eux comme le montre la figure ci-dessous :

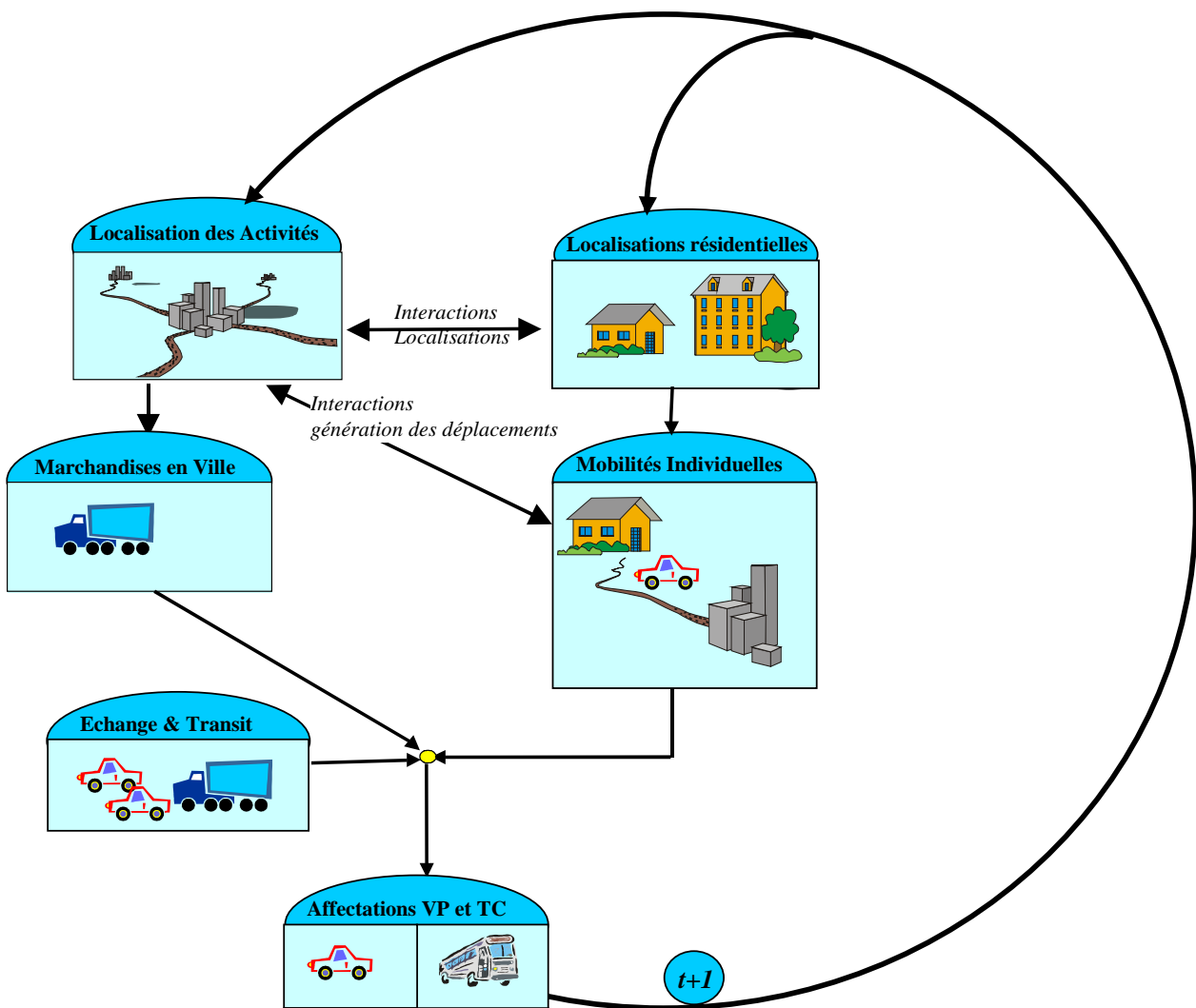


Figure 2-2 : l'architecture des 6 modules du projet SIMBAD (NICOLAS et alii, Rapport n°4 : l'architecture du modèle au sein du projet SIMBAD, Mars 2006)

- Les modules de localisation :
 - Des activités économiques (emplois et établissements)
 - Des résidences des ménages de l'aire étudiée
- Les modules de génération et de répartition (horaire et modale) des trafics
 - De personnes
 - De marchandises
 - D'échange et de transit
- Et enfin le module de l'affectation des véhicules particuliers sur la voirie et des utilisateurs des transports collectifs.

Cette architecture décomposée en 6 modules comme il est présenté sur la Figure 2-2 repose sur la structure du modèle intégré simulant les interactions transport et urbanisme UrbanSim. Il est à la base notamment de la boucle de retour des affectations VP et TC aux modules de localisation ce qui schématise l'influence en retour des infrastructures de transport sur l'urbanisme et l'occupation des sols. Influence qui joue un rôle dans un horizon à long terme qui justifie la volonté d'établir des simulations jusqu'à 2025 sur l'aire urbaine lyonnaise.

Voyons à présent comment s'organise le modèle sur lequel s'appuie le projet global SIMBAD.

2.3 Le modèle Urbansim

2.3.1 Introduction

Le modèle Urbansim est apparu à la fin des années 90 aux Etats-Unis(1999) avec à la prise en compte des interactions existant entre développement urbain, transport et environnement. Le fait par exemple que la mise en place d'une voie rapide urbaine engendre non seulement de nouveaux déplacements mais aussi une augmentation des émissions de gaz ainsi qu'un nouveau développement de l'espace urbain (nouvelles localisations de résidences ou d'activités économiques) a suscité de nouvelles interrogations dans les politiques de transport et de planification urbaine. L'appui de cette prise de conscience par la législation avec les lois sur l'air ISTEA (Intermodal Surface Transportation Efficiency) en 1991 suivie du TEA21 (Transportation Equity Act for the 21st Century) en 1998 ont amené les Metropolitan Planning Organizations (MPOs) à concilier leurs travaux en matière de planification urbaine et des transports avec une attention particulière portée à la problématique environnementale (The UrbanSim Project, Juillet 2007, www.urbansim.org).

A partir de ce moment là, de nouvelles recherches en matière de modélisation de trafics et plus généralement des transports ont été lancées. En effet, les précédents modèles développés reposaient principalement sur les problématiques de congestion ou de trafic. Ils ne prenaient alors pas en compte les différentes interactions entre transport, urbanisation et environnement.

Ainsi, au milieu des années 1990, en réponse au manque de modèles traitant les interactions transport et urbanisme plusieurs projets naissent dont le Travel Model Improvement Project (TMIP) commun aux Administrations Fédérales de Transit et des Routes ainsi qu'à l'Agence de Protection de l'Aménagement. Rapidement il devient le Travel and Land Use Model Improvement Project (TLUMIP) mené notamment par le département de transport d'Oregon. A partir de 1996, ce projet s'attache à développer de nouveaux modèles intégrés permettant l'évaluation des interactions entre transport et urbanisme dont Urbansim. Les premiers travaux sont donc été effectués sur l'aire métropolitaine d'Eugene-Springfield pour être ensuite appliqués à Honolulu, Hawaii et Salt Lake City.

Urbansim est une plateforme créée en 1999 visant à considérer les problématiques environnementales, sociales et économiques dans des projets de planification urbaine. Il met en avant au départ les relations qui peuvent exister entre les choix des agents économiques (ménages, entreprises) vis à vis de l'évolution du marché de l'immobilier.

2.3.2 Présentation de l'architecture

Urbansim comprend plusieurs entrées exogènes. Deux d'entre elles sont issues de systèmes modèles externes : un modèle macroéconomique prévoyant les évolutions macroéconomiques futures comme pour la population et l'emploi par secteur, et un modèle de demande de transport prévoyant les conditions de transport comme les temps de congestion et les utilités composées d'un trajet entre chaque échange. Ce dernier est très lié à UrbanSim, avec la prise en compte de l'utilisation du sol par le modèle de transport externe, et la prise en compte des conditions de transport par le modèle d'occupation des sols d'UrbanSim au cours de l'itération de l'année suivante.

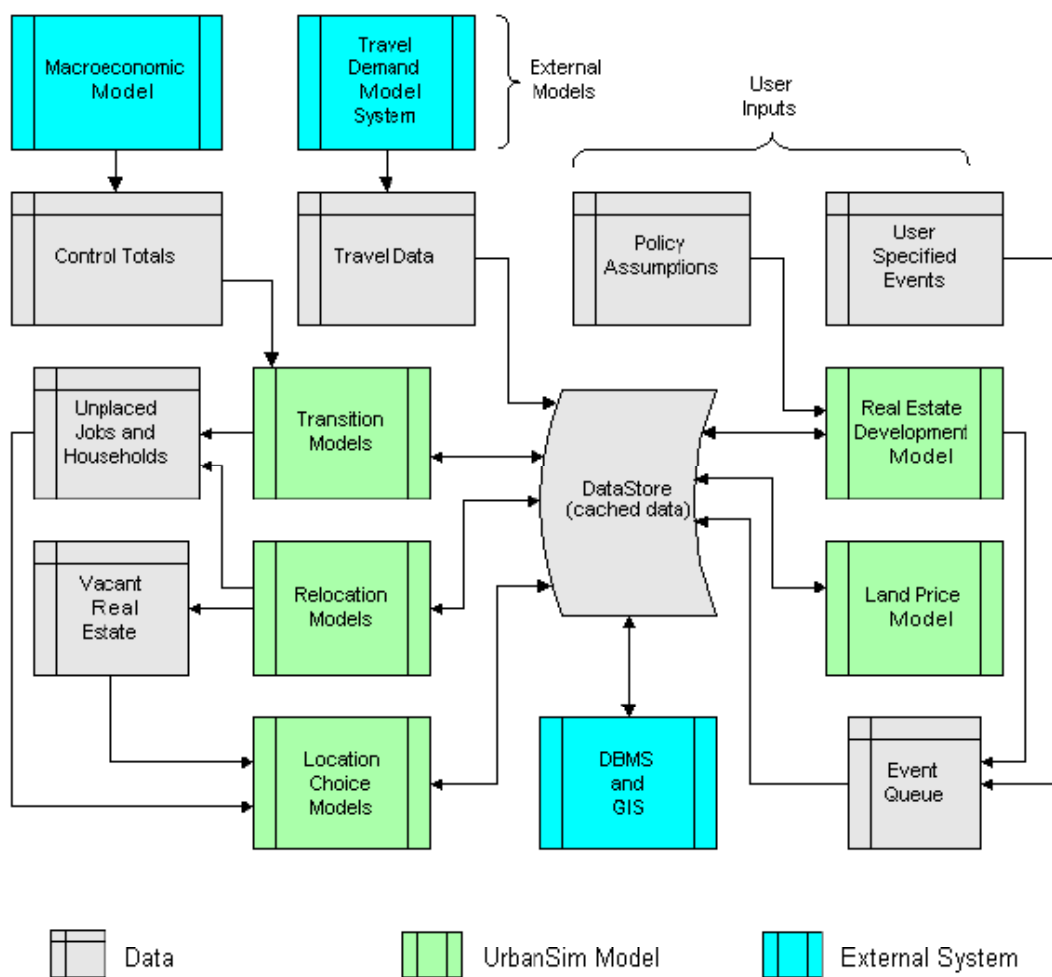


Figure 2-2 : La structure du modèle Urbansim (source :The UrbanSim Project, Juillet 2007)

UrbanSim fonctionne sur une base annuelle de simulation, et le flux de données est comme indiqué sur la Figure 2-2. Le dossier de stockage de données (DataStore) contient l'état en cours de tous les objets du système. Chacun des principaux modules est décrit dans les paragraphes suivants.

Le modèle lit des données d'entrée exogènes non seulement venant des modèles externes macroéconomiques et de demande de transport, mais aussi venant de l'utilisateur d'UrbanSim. Il lui est possible d'entrer des événements spécifiques (« User Specified Events ») qui, selon ses hypothèses (avec pour exemple la mise en place de politiques de régulation des transports par le stationnement, ou d'une hausse des prix du foncier dans le centre-ville), auront une influence sur l'occupation des sols, l'emploi ou l'organisation des transports urbains.

Les composants du modèle principal, dans l'ordre d'exécution sont les modèles de transition économique et démographique, les modèles de mobilité des ménages et des emplois, le modèle d'accessibilité, les modèles de localisation des ménages et des emplois, le modèle de développement immobilier et le modèle de détermination du prix du foncier. Décrivons de manière plus précise ces différents modules.

2.3.2.1 *Le module de transition*

Ce module est composé de deux modèles de transition comme nous l'avons évoqué plus haut : l'un concernant l'économie, l'autre la démographie.

Le modèle de transition économique (« economic transition model ») :

Ce modèle est classifié suivant l'appartenance à un secteur d'activité. S'appuyant sur la structure de l'économie locale, UrbanSim est capable de définir 20 secteurs d'activité distincts. Les prévisions agrégées de l'activité et de l'emploi sont exogènes à UrbanSim. Ces prévisions sont prises en compte par UrbanSim dans le modèle de transition économique qui permet de déterminer le taux de croissance ou de diminution de l'emploi par secteur d'activité sur l'année précédente, pour ainsi créer ou détruire des emplois. Ces emplois, dans le cas des créations, seront chargés dans la base de données d'entrée au modèle de localisation en attente d'affectation. Concernant les emplois détruits, l'espace libéré est déclaré disponible dans la base de données du modèle de localisation des emplois.

Rappelons par ailleurs que dans le cadre de SIMBAD, l'unité économique ne sera pas l'emploi mais l'établissement. Un travail que nous n'allons pas aborder dans cet exposé est donc mené parallèlement afin de permettre une bonne prise en compte de ce changement de perception.

Le modèle de transition démographique (« demographic transition model ») :

Ce modèle est basé sur la même structure et la même dynamique que le modèle précédent. Il détermine les modifications de la distribution des ménages suivant leur type. En réalité, ces modifications sont le résultat d'un ensemble complexe de changements démographiques et sociaux comme le vieillissement de la population, la constitution du ménage, le divorce, la mortalité, la naissance d'enfants, le mouvement des ménages dans la région, ou encore le revenu des ménages. Ces modifications sont également intégrées au moyen de données exogènes au module.

Comme dans le modèle de transition économique, les ménages créés sont stockés dans une base de données en attente de localisation établie par le modèle de localisation démographique. Enfin de la même manière, l'espace libéré par le ménage détruit est rendu disponible.

2.3.2.2 *Le module de mobilité*

A l'image du module de transition, le module de mobilité est constitué de deux modèles : l'un de l'emploi, l'autre du ménage.

Le modèle de mobilité économique (« employment relocation model ») :

La délocalisation des emplois est une décision prise par l'établissement. L'unité d'analyse choisie est l'emploi individuel et non pas l'établissement comme cela sera fait dans SIMBAD. Le module détermine la probabilité que chaque emploi soit déplacé d'une cellule à l'autre, à l'issue d'une année t . Ces transitions reflètent la conjoncture sur le marché du travail. Les emplois ainsi déplacés seront relocalisés par le module de localisation (« employment location choice model »). L'espace libéré est déclaré vacant dans la base de données. Bien qu'il semble naturel que la décision de mobilité soit conditionnelle à l'espace de relocalisation (modèle imbriqué, conditionnel à la localisation), cela n'est pas pris en compte ici. En effet, cela pose des problèmes en termes de calage des données. Le choix de mobilité est considéré comme indépendant du choix de localisation. Le modèle de

mobilité de l'emploi fonctionne alors de la façon suivante : l'ensemble des emplois déplacés (Mst) est déterminé par tirage aléatoire dans l'ensemble Jat, à l'aide du taux de mobilité exogène dans le secteur s.

Le modèle de mobilité démographique (« household relocation model ») :

La mobilité des ménages est basée sur la même structure et la même dynamique que celle des emplois présentée dans le paragraphe précédent. A partir des données démographiques, des taux exogènes de mobilité par catégorie de ménage sont introduits. Ces données (probabilité de déménagement) issues de l'enquête Origine-Destination de 2002 sur l'aire urbaine Lyonnaise sont en cours de traitement.

2.3.2.3 Le module d'accessibilité

De nombreuses études ont montré l'importance de l'accessibilité dans le choix de localisation résidentielle des ménages et des entreprises. La notion d'attractivité d'une zone permet de lui attribuer une valeur économique. La zone est ici une zone du découpage du modèle de trafic. L'attractivité est considérée comme un bien normal (dont la demande augmente avec le revenu), comme tout autre attribut du logement. Les individus attribuent une plus ou moins grande valeur à chaque localisation selon qu'il permet un accès plus ou moins aisé aux emplois et aux zones d'activités. On considère également que la demande pour ce bien varie selon les individus. Par exemple, les retraités n'ont pas le même intérêt pour une zone proche des emplois que les individus actifs.

$$A_i = \sum_{j=1}^J D_j e^{L_{a;j}}$$

Où :

- D_j est le volume d'activité de la zone j (nombre d'emplois ou de surfaces commerciales par exemple)
- $L_{a;j}$ est la Logsomme des ménages de niveau a de motorisation et qui se déplacent de la zone i vers la zone j
- J , le nombre total de zones.

Ce calcul est effectué chaque année, à l'aide de la matrice issue du modèle de trafic et des données d'occupation du sol. Cela permet ensuite de créer des indicateurs d'accessibilité qui seront utilisés dans les modules de choix de localisation. Il n'y a donc pas d'interaction intra annuelle entre transport et occupation du sol puisque en année t , l'attractivité est calculée en utilisant les données de l'année précédente ($t-1$).

2.3.2.4 *Le module de localisation*

Le modèle de localisation des emplois (« employment location choice model ») :

Ce module calcule la probabilité qu'un nouvel emploi (créé dans le modèle de transition économique) ou relocalisé (mobilité économique) soit localisé dans une cellule donnée. Chaque cellule contient un espace donné qui peut être occupé par des emplois. Cet espace total disponible pour l'emploi dépend de l'espace non résidentiel et de la densité d'usage. Il est également possible de localiser des emplois dans des zones résidentielles.

Dans ce module de localisation des emplois (ainsi que pour la localisation des ménages), le stock d'espace disponible est fixé dans la période intra annuelle de simulation.

Parmi les variables pouvant influencer le choix de localisation, on trouve :

- l'accessibilité
- l'âge des immeubles
- la densité des infrastructures
- la proximité d'un axe de voirie
- le niveau de desserte de la zone en transports en commun : desserte spatiale (présence d'une gare), desserte temporelle (fréquence)
- les prix.

L'ensemble de ces variables explicatives est introduit dans un modèle multinomial Logit afin de déterminer la probabilité qu'un emploi (placé aléatoirement dans une file d'attente) se localise à un endroit donné. Une fois cet emploi localisé, l'espace correspondant est supprimé et le module itère le processus pour l'emploi suivant en attente.

Le modèle de localisation des ménages (« household location choice model ») :

Ce modèle a une structure similaire à celle du modèle précédent. Grâce à un modèle logit multinomial, ce modèle détermine la probabilité qu'un ménage nouveau ou déplacé se localise dans une cellule donnée. Il est possible de considérer différents types de ménages, selon le revenu, le nombre d'enfants ou la catégorie socioprofessionnelle.

Les variables explicatives, caractéristiques de la cellule, pourraient être :

- le coût du transport et du terrain (reflétant l'arbitrage entre les deux biens dans le modèle monocentrique)
- la densité urbaine
- le revenu moyen de la zone (effet d'imitation)
- les caractéristiques du logement (en particulier la disponibilité de places de stationnement privé) et de l'environnement
- le niveau de desserte de la zone en transports en commun : desserte spatiale (présence d'une gare), desserte temporelle (fréquence)

2.3.2.5 *Le module de développement immobilier*

Ce module détermine la probabilité de réalisation de programmes immobiliers, que ce soit de la création ou de la réhabilitation. Chaque cellule, selon sa superficie affectée aux espaces non résidentiels et selon le nombre d'unités d'habitations, appartient à un type de développement (ou type urbain) donné.

Les données nécessaires à l'estimation du module de développement sont issues du calcul, pour chaque cellule et pour chaque année de son historique, de l'ensemble des transitions entre types de cellules. Le module peut ensuite estimer, à l'aide d'un modèle multinomial Logit, la probabilité qu'une cellule modifie son type (c'est à dire qu'un programme immobilier permettant de passer d'un type à un autre se réalise). Ce changement est ensuite déterminé par un tirage aléatoire. La modification éventuelle est introduite dans la base de données. Un certain nombre d'éléments peuvent venir contraindre les changements de type : plans local d'urbanisme, contraintes physiques, techniques ou environnementales.

Les variables explicatives rendant compte du choix de type urbain sont les suivantes :

- les caractéristiques du site : programme de développement en cours, PLU, ...
- caractéristiques urbaines,
- maillage routier,
- proximité d'un programme immobilier,
- espaces verts,
- caractéristiques régionales,
- temps de parcours vers les aéroports, vers des pôles d'emplois tels que la Part-Dieu.
- accessibilité,
- caractéristiques du marché,
- taux de surfaces résidentielles,
- prix du logement

A noter que les échelles de temps entre le modèle de développement immobilier et les modèles de localisation des ménages et des emplois sont très différentes. Le premier peut s'inscrire sur des durées de l'ordre d'une vingtaine d'années alors que l'on est sur des bases inférieures à la décennie pour les deux autres.

2.3.2.6 *Le module de détermination du prix du foncier*

Les prix sont des indicateurs de l'adéquation entre l'offre et la demande de terrain, entre les différentes zones et les différents types. Les prix expliquent en partie les choix de localisation des entreprises et des ménages. Evidemment, il y aura toujours des sociétés qui recherchent des adresses de prestige pour leur « image » sans regarder les prix. Ce sont des cas particuliers non modélisables. Mais généralement, des prix trop élevés en centre-ville poussent les entreprises à chercher de nouveaux locaux en périphérie.

Parmi les variables à prendre en compte dans le module, il y a les variables d'accessibilité et de desserte. On sait que l'implantation d'une nouvelle gare ou station de transports en commun lourds augmente significativement les prix de l'immobilier dans les environs proches.

Les hypothèses suivantes sont prises :

- L'ensemble des agents est preneur de prix et les ajustements sont faits en réponse à la demande et l'offre agrégées.
- Les préférences et les déséquilibres entre offre et demande se reportent dans la valeur du terrain. La valeur d'un immeuble reflète uniquement son coût de remplacement.
- Il existe un taux de vacance structurel de long terme pour chaque type de propriété et la relation entre le taux de court terme et le taux de long terme influence le prix des propriétés.

3 ANALYSE TECHNIQUE, GUIDE D'UTILISATEUR DU MODELE URBANSIM.

Après avoir présenté l'architecture globale d'Urbansim avec ses différents modules et les modèles qu'il met en jeu, nous allons apporter des précisions et des éclaircissements à cette architecture dans le but de faciliter son utilisation et sa compréhension. Cette réflexion autour de l'organisation d'Urbansim a été appuyée par le fait qu'à moyen terme, comme nous l'avons présenté dans la première partie, Urbansim servira d'architecture de base pour le projet SIMBAD mis en place sur l'aire urbaine lyonnaise. De ce fait, il devient nécessaire de repérer les points importants du modèle afin de pouvoir les prendre en compte et de pouvoir envisager les éventuelles modifications ou ajustements qu'il faudrait effectuer lors des modélisations sur l'aire et les données lyonnaises. Ce travail de description et de repérage des éléments essentiels se pose comme le point de départ du travail qui est fait sur Urbansim dans le Laboratoire d'Economie des Transports avec :

- dans un premier temps, les premiers tests effectués sur la base des données de la ville d'Eugene-Springfield mise à disposition librement sur le site www.urbansim.org, tests menés parallèlement aux travaux de recueils et de premiers traitements des données d'entrée Lyonnaises (données sur les ménages, les activités économiques avec le traitement des enquêtes-ménages et des données INSEE à disposition) ainsi qu'à ceux concernant les modèles de transport DAVISUM et FRETURB,
- ces travaux se poursuivront dans un deuxième temps avec l'intégration des données lyonnaises au modèle Urbansim.

3.1 Le travail sur les données

3.1.1 Les données d'entrée

Après avoir installé Urbansim (version 4.0) en respectant les différentes étapes et effectué les différents tests comme il est recommandé par le tutorial présenté www.urbansim.org, nous avons cherché à nous intéresser aux données d'entrée disponibles.

Les données d'entrée sont de quatre types :

- Données de base du modèle
- Constante
- Données de scénario
- Paramètres des modèles

Nom du fichier de données	Type du fichier
'development_events'	Scénario
'development_constraints'	Scénario

'target_vacancies'	<i>Scénario</i>
'land_use_events'	<i>Scénario</i>
'employment_events'	<i>Scénario</i>
'annual_employment_control_totals'	<i>Données</i>
'annual_household_control_totals'	<i>Données</i>
'buildings'	<i>Données</i>
'building_types'	<i>Données</i>
'target_vacancies'	<i>Données</i>
'gridcells'	<i>Données</i>
'households'	<i>Données</i>
'job_building_types'	<i>Données</i>
'jobs'	<i>Données</i>
'travel_data'	<i>Données</i>
'zones'	<i>Données</i>
'household_characteristics_for_ht'	<i>Données</i>
'commercial_development_location_choice_model_coefficients'	<i>Paramètres</i>
'commercial_development_location_choice_model_specification'	
'commercial_employment_location_choice_model_coefficients'	<i>Paramètres</i>
'commercial_employment_location_choice_model_specification'	
'home_based_employment_location_choice_model_coefficients'	<i>Paramètres</i>
'home_based_employment_location_choice_model_specification'	
'industrial_employment_location_choice_model_coefficients'	<i>Paramètres</i>
'industrial_employment_location_choice_model_specification'	
'industrial_development_location_choice_model_coefficients'	<i>Paramètres</i>
'industrial_development_location_choice_model_specification'	
'residential_development_location_choice_model_coefficients'	<i>Paramètres</i>
'residential_development_location_choice_model_specification'	
'household_location_choice_model_coefficients'	<i>Paramètres</i>
'household_location_choice_model_specification'	
'land_price_model_coefficients'	<i>Paramètres</i>
'land_price_model_specification'	
'residential_land_share_model_coefficients'	<i>Paramètres</i>
'residential_land_share_model_specification'	
'annual_relocation_rates_for_households'	<i>Paramètres</i>
'annual_relocation_rates_for_jobs'	<i>Paramètres</i>
'counties'	<i>Constantes</i>
'urbansim_constants'	<i>Constantes</i>

'development_types'	<i>Constantes</i>
'development_type_group_definitions'	<i>Constantes</i>
'base_year'	<i>Constantes</i>
'cities'	<i>Constantes</i>
'development_type_groups'	<i>Constantes</i>
'employment_adhoc_sector_group_definitions'	<i>Constantes</i>
'employment_adhoc_sector_groups'	<i>Constantes</i>
'employment_sectors'	<i>Constantes</i>
'plan_types'	<i>Constantes</i>
'race_names'	<i>Constantes</i>

Tableau 1: Liste des fichiers de données d'entrée.

3.1.2 L'affichage des tables

Le premier problème qui se pose concernant toutes ces données stockées dans le fichier « eugene_1980_baseyear_cache » mis à disposition sur le site www.urbansim.org est celui de l'affichage et du format de lecture. En effet, ces données sont fournies sous forme de dossiers comprenant plusieurs fichiers au format .li4 et non de tables. Prenons l'exemple de la table « gridcells » dont la structure d'une cellule est présentée par le tableau ci-dessous.

Champ	Type de données	Description
Grid_id	<i>Entier</i>	Identifiant unique
Relative_x	<i>Entier</i>	Coordonnee x dans le systeme de coordonnees de la grille
Relative_y	<i>Entier</i>	Coordonnee y dans le systeme de coordonnees de la grille
City_id	<i>Entier</i>	Ville à laquelle appartient cette cellule
County_id	<i>Entier</i>	Comte auquel appartient cette cellule
Zone_id	<i>Entier</i>	Zone d'analyse de trafic que contient le centroïde ² de cette cellule
Year_built	<i>Entier</i>	Ex : 2002

² Centroïde : Point fictif situé à l'intérieur d'un polygone et dont les coordonnées correspondent approximativement au centre de ce polygone (Source : www.mrn.gouv.qc.ca/territoire/geomatique/geomatique-vocabulaire.jsp)

Nonresidential_land_value	<i>Entier</i>	Unite, ex : dollars
Residential_land_value	<i>Entier</i>	Unite, ex : dollars
Percent_water	<i>Entier</i>	Pourcentage de cette cellule couvert par des rivières ou cours d'eau
Percent_wetland	<i>Entier</i>	Pourcentage de cette cellule couvert par des zones humides ou marecageuses
Percent_slope	<i>Entier</i>	Pourcentage de cette cellule en des devers
Percent_public_space	<i>Entier</i>	Pourcentage de cette cellule couvert par des espaces publics
Percent_roads	<i>Entier</i>	Pourcentage de cette cellule couvert par des routes
Percent_stream_buffer	<i>Entier</i>	
Percent_floodplain	<i>Entier</i>	
Percent_open_space	<i>Entier</i>	
Commercial_sqft	<i>Entier</i>	
Developmetn_type_id	<i>Entier</i>	
Distance_to_arterial	<i>Flottant</i>	
Distance_to_highway	<i>Flottant</i>	
Governmental_sqft	<i>Entier</i>	
Industrial_sqft	<i>Entier</i>	
Commercial_improvement_value	<i>Entier</i>	
Industrial_improvement_value	<i>Entier</i>	
Governmental_improvement_value	<i>Entier</i>	
Residential_improvement_value	<i>Entier</i>	
Residential_units	<i>Entier</i>	
Plan_type_id	<i>Entier</i>	
Is_outside_urban_growth_boundary	<i>Flottant</i>	
Fraction_residential_land	<i>Flottant</i>	

Tableau 2 : Caractéristiques d'une cellule de la table "gridcells".

Cette table comprend des informations repérées géographiquement par rapport à chaque cellule de la grille quadrillant la zone d'étude. Ce fichier comporte pour chaque cellule du maillage de l'année initiale les correspondances avec d'autres découpages, des attributs d'occupation du sol, des valeurs foncières, des contraintes environnementales. Elle apparaît dans le fichier « eugene_1980_baseyear_cache » comme un dossier comprenant 31 fichiers (à des formats .li4, .lf4 ou .lf8 qui ne sont pas lisibles directement) correspondant aux 31 colonnes de la table « gridcells ». Ainsi, nous avons cherché à extraire ses données afin de pouvoir les visualiser puis afin de chercher à les modifier dans l'optique de travailler sur les données lyonnaises. Urbansim permet cela à l'aide

du script « export_cache_to_mysql.py » (voir ci-dessous) situé dans Python24\Lib\site-packages\eugene\tools :

```
import os
import sys

from opus_core.tools.gui_export_cache_to_mysql_database import __file__ as
script_path

os.system('%s %s' % (sys.executable, script_path) )
```

Les données sont ainsi exportées vers le logiciel de traitement de données MySQL qui correspond vraiment au traitement de ce type de données. En effet, il existe également des possibilités d'exporter les données espacées par des tabulations ou des virgules sous format texte, format qui peut être lu par des logiciels comme Excel. Cependant la taille des données devient vite contraignant Excel ne pouvant seulement recevoir par exemple 65536 lignes, ce qui s'avère insuffisant dans notre cas et qui pourra poser problème dans l'exploitation ou la représentation des données. Les tables sont alors ainsi chargées dans le répertoire « Data » de MySQL et apparaissent sous forme de tableau comme présenté dans l'image suivante :

residen...	residential_units	year_built	zone_id	percent_floodpl...	distance_to_art...	relative_y	relative_x	commercial_sqft
0	0	0	26	0	150	81	218	0
000071...	0	0	28	0	150	91	178	0
-1	22	1970	161	0	150	53	46	679
1	35	1970	276	0	600	61	117	0
-1	10	1944	169	0	0	81	141	37258
1	18	1949	71	0	1530	91	101	0
1	0	0	165	0	3092	33	8	0
0	0	0	35	0	2163	111	152	0
0	0	0	35	0	2620	104	146	0
1	8	1968	210	0	750	83	64	0
-1	64	1963	174	0	0	70	129	73
-1	9	1970	161	0	450	55	43	18059
1	26	1966	121	0	2089	91	77	0
0	0	1977	152	0	900	71	53	1773

Figure 3-1: Illustration de la visualisation d'un extrait de la table "gridcells" après extraction vers le logiciel MySQL.

3.1.3 Les données de sortie

Les données de sortie disponibles après simulation sont stockées dans un répertoire automatiquement créé avec chaque simulation effectuée. Il comprend les tables présentées dans le tableau ci-dessous dans lequel on retrouve les données de scénario (« Scénario ») et les données de base du modèle modifiées (« Données ») après la simulation qui aura appelée les différents modèles. Pour ce qui est des paramètres des modèles ils sont inchangés car ces paramètres sont établis au cours de l'estimation des modèles qui a lieu avant la simulation ce que nous présenterons par la suite. Les constantes sont également inchangées et donc absentes de ces données de sortie.

Nom du fichier de données	Type du fichier
'development_events'	<i>Scénario</i>
'development_constraints'	<i>Scénario</i>
'target_vacancies'	<i>Scénario</i>
'land_use_events'	<i>Scénario</i>
'employment_events'	<i>Scénario</i>
'all_data'	<i>Données</i>
'building_types'	<i>Données</i>
'target_vacancies'	<i>Données</i>
'gridcells'	<i>Données</i>
'households'	<i>Données</i>
'job_building_types'	<i>Données</i>
'jobs'	<i>Données</i>
'travel_data'	<i>Données</i>
'zones'	<i>Données</i>
'development_types'	<i>Constantes</i>
'development_type_group_definitions'	<i>Constantes</i>
'development_type_groups'	<i>Constantes</i>

Tableau 3: Liste des fichiers de données de sortie

3.1.4 Les représentations des données

Après avoir réussi à obtenir un affichage sous forme de tableau des données d'Eugene-Springfield à l'aide du logiciel de traitement de données MySQL qui nous a permis d'analyser et de juger les modifications des valeurs au cours des différentes simulations, nous avons cherché à illustrer ces tables par des représentations graphiques et cartographiques pouvant appuyé ces changements.

Ainsi notre intérêt s'est porté vers la table « gridcells » présenté un peu plus haut qui comprend les champs de localisation géographique (coordonnées x et y du système de la grille voir Tableau 2) permettant le repérage des différentes cellules. Ainsi cette table comprenant ces informations géographiques, ajoutées aux champs donnant des informations au niveau de l'occupation des sols ou du foncier, peut être utilisée avec des logiciels SIG (Systèmes d'Information Géographique) comme le logiciel gratuit gvSIG qui fournit des images présentées ci-dessous.

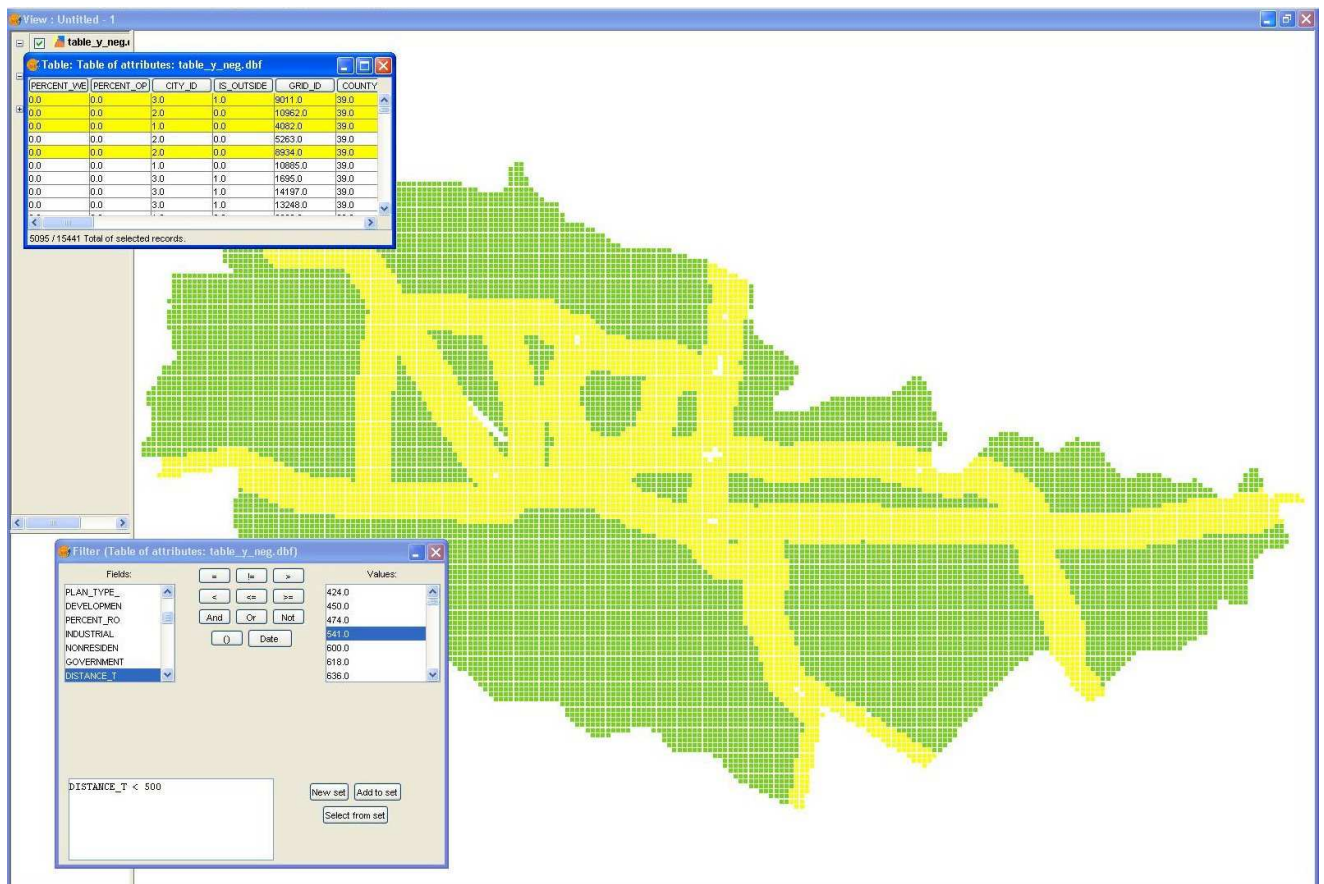


Figure 3-2 : Représentation des cellules à moins de 500m d'une artère routière.

Ces cartes permettent ainsi de représenter et d'illustrer les évolutions des données et des caractéristiques des cellules au cours des différentes années simulées. Dans l'optique de travailler sur les données lyonnaises, il sera intéressant et primordial d'utiliser ces cartes afin de pouvoir noter l'influence des différents indicateurs et donc de pouvoir évaluer les choix à retenir. Par ailleurs, d'un point de vue technique, gvSIG présente l'avantage d'être directement relié au logiciel de gestion de données MySQL. Les transferts et l'affichage des données se font rapidement et ne nécessitent aucune conversion.

Enfin comme le souligne la Figure 3-3, il offre également la possibilité d'effectuer des requêtes en croisant plusieurs champs selon le phénomène ou la caractéristique que l'auteur cherche à révéler.

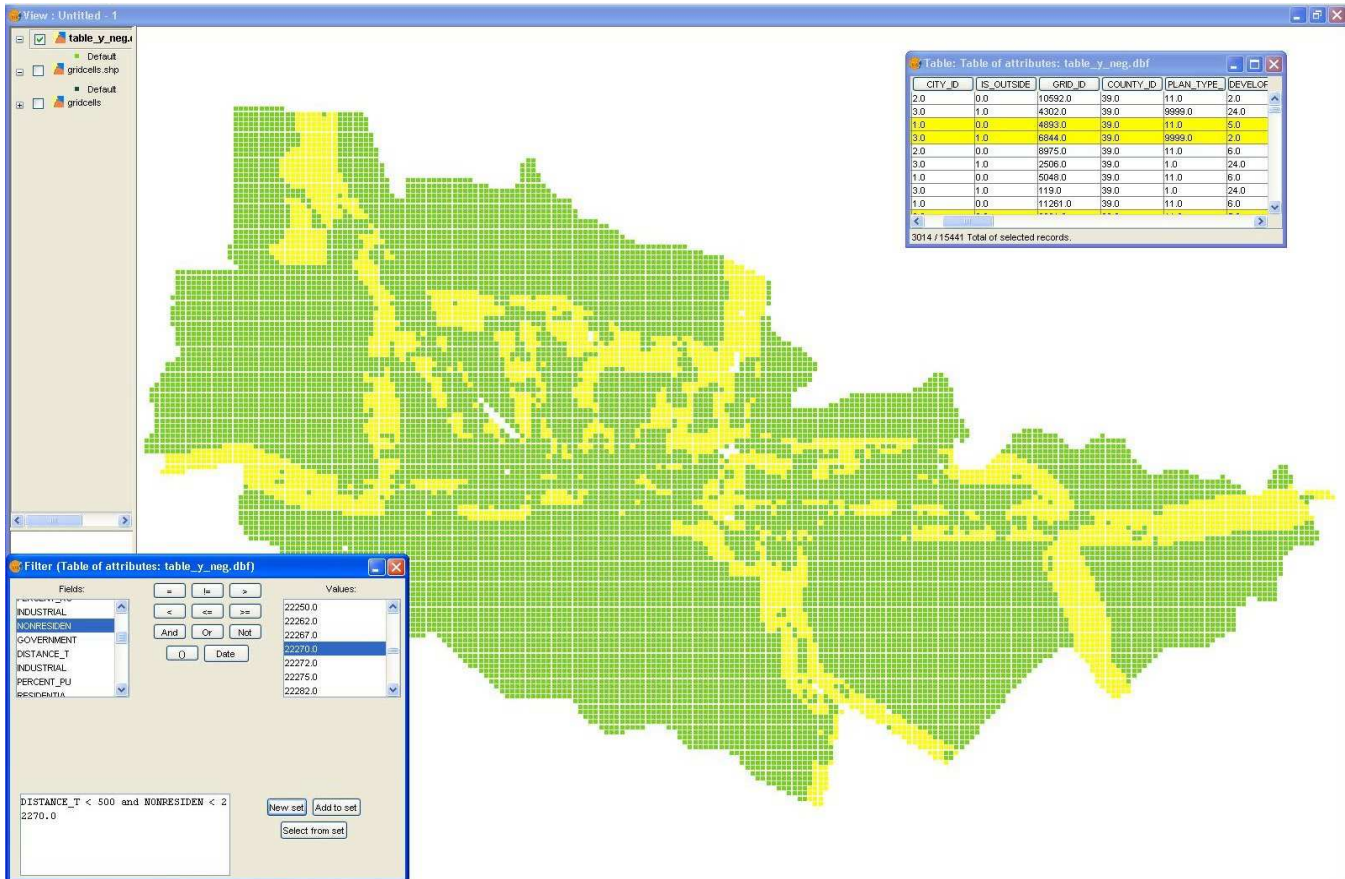


Figure 3-3 : Représentation des cellules distantes de moins de 500m d'une artère routière et dont la valeur non résidentielle est inférieure à 22270\$.

3.2 Lancer une simulation

Après avoir travaillé sur les données disponibles, leur lecture, leur affichage et leur représentation, la volonté a été d'apporter des éclaircissements sur l'organisation et la mise en place du modèle. Nous allons ainsi dans cette partie présenter les éléments importants de modélisation nécessaires au bon fonctionnement d'Urbansim.

3.2.1 Le codage utilisé

Urbansim est un modèle utilisant l'interface de codage Python. Il s'organise en une succession importante de scripts appelant les modèles à mettre en œuvre, les données s'y référant ou encore les spécifications et coefficients des modèles. Ces scripts se trouvent dans les différents site-packages du dossier Python 24 installé suivant les descriptions faites sur www.urbansim.org comme par exemple :

- C:\Python24\Lib\site-packages\eugene\tools
- C:\Python24\Lib\site-packages\eugene\configs
- C:\Python24\Lib\site-packages\opus_core\configurations
- C:\Python24\Lib\site-packages\urbansim\configs

Cette diversité dans les trajets pour obtenir les différents scripts ne facilitant pas la prise en main et la lecture de l'organisation globale du modèle, nous avons utilisé l'interface du logiciel Eclipse qui permet une meilleure lecture de ces scripts. Un effort de recherche et de repérage des scripts importants a donc été fait afin de bien faire apparaître l'architecture et les enchaînements du modèle. Nous les avons ensuite regroupé et organisé en arborescence dans différents dossiers d'Eclipse avec pour dossier principal MyNew_Eugene :

MyNew_Eugene :

- Estimation : dossier comprenant tous les scripts nécessaires à l'estimation des coefficients modèles
- Tools : dossier comprenant notamment le script lançant la simulation, ou encore celui permettant d'extraire les données vers MySQL
- Configs : dossier comprenant le script décrivant les étapes de la simulation (les modèles appelés, les tables à prendre en compte)
- Models : dossier comprenant les scripts des modèles appelés dans la simulation

3.2.2 Etape préalable à la simulation, l'estimation des modèles

Avant de lancer une simulation, il est nécessaire d'accomplir une étape préliminaire indispensable : l'estimation des paramètres des modèles. Cette étape est indépendante de la procédure globale de la simulation et doit être accomplie en amont. Une fois effectuée, les coefficients et spécifications de tous les modèles sont estimés et fixés, les simulations peuvent alors être lancées.

Nous allons nous intéresser à cette étape en nous appuyant sur l'exemple du modèle Land Price Model (modèle de détermination du prix du foncier) sur lequel nous avons accompli les premiers tests et modifications. Chaque modèle utilisé dans Urbansim est basé sur la même organisation :

- Un script `name_model.py` (`land_price_model.py` dans notre exemple) l'identifiant et précisant ses caractéristiques et les scripts qu'il appelle, à savoir le modèle sur lequel il se base (un modèle de régression hédonique dans notre cas, `regression_model.py`).
- Chaque modèle est spécifié par un ensemble de variables. Celles-ci sont connectées à un ensemble de coefficients calibrés sur les données observées. Ces spécifications et coefficients sont précisés dans les tables `name_specifications` (`land_price_model_specifications`) et `name_coefficients` (`land_price_model_coefficients`) présentes dans les données d'entrée (voir Tableau 1). Dans notre exemple ces données d'entrée seront nommées « Expbase1980 » qui n'est autre qu'une copie de « `eugene_1980_baseyear_cache` ».
- Enfin, pour obtenir ces tables de spécifications et coefficients (ie les paramètres du modèle) nécessaires aux simulations, un script est indispensable : `INITIALESDUMODELE_specifications.py` (`LPM_specifications.py`). Il spécifie les variables ou « spécifications » sur lesquelles sont calibrés les coefficients.

Nous venons ainsi de décrire rapidement l'architecture concernant l'estimation d'un modèle. Intéressons nous maintenant aux scripts à l'origine de cette estimation, c'est à dire la base de la partie estimation d'Urbansim.

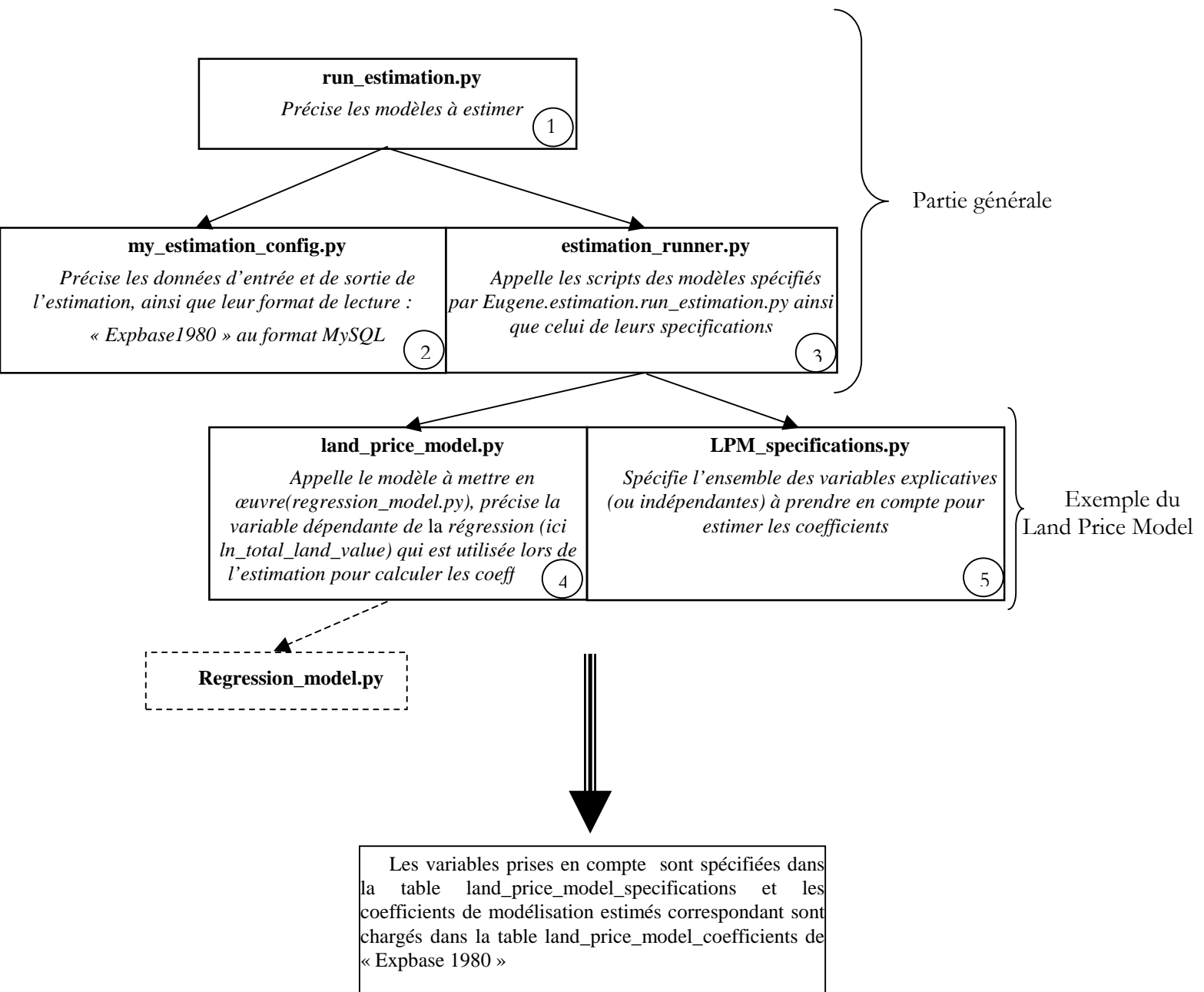


Figure 3-4 : Architecture et enchaînement des scripts de l'estimation

Revenons maintenant plus dans le détail sur chaque étape et chaque script nécessaire à la réalisation de l'estimation des paramètres afin d'apporter des précisions et de relever les éléments importants qui pourront jouer un rôle dans la future prise en compte des données lyonnaises. En effet, ce travail en amont permet de repérer les éventuelles modifications qui pourront être apportées au cours du travail sur l'aire urbaine lyonnaise, aussi bien au niveau des données, des indicateurs que des paramètres de modélisation.

1. ***run_estimation.py*** (situé dans EugTest\eugene\estimation dans Eclipse, C:\MyNew\projects\OPUS\opus-4.0\eugene\estimation)

est à l'origine de la réalisation de l'estimation. Une fois que les modifications sont effectuées, que les choix de bases de données, de variables et de paramètres sont spécifiés dans les cinq scripts présentés ci-dessus, l'estimation est lancée à l'aide de ce script. Il précise les différents modèles à estimer comme le montre l'extrait du script ci-dessous.

```
from opus_core.storage_factory import StorageFactory

#model = 'hlcm'
#model = 'elcm-industrial'
#model = 'elcm-commercial'
#model = 'elcm-home_based'
#model = 'dplcm-industrial'
#model = 'dplcm-commercial'
#model = 'dplcm-residential'
model = 'lpm'
#model = 'rlsm'
```

Cet extrait du code correspond au choix des modèles à estimer. Nous avons choisi dans notre cas de travailler seulement sur le modèle Land Price Mode appelé '*lpm*'. L'estimation des paramètres des autres modèles ne se fera pas car ils ne sont pas appelés par ce script du fait des # situés devant la ligne de codage leur correspondant (au contraire du Land Price Model).

2. ***my_estimation_config.py*** (situé dans C:\MyNew\projects\OPUS\opus-4.0\eugene\estimation ou EugTest\eugene\estimation dans Eclipse)

informe des configurations générales de l'estimation, à savoir le choix des données utilisées en entrée, l'endroit où elles sont stockées en sortie après estimation et enfin le format sous lequel elles sont lues.

Dans notre cas, comme nous l'avons présenté plus haut, le choix a été de travailler avec MySQL afin de faciliter la lecture et le traitement des données. La base d'entrée se nomme « Expbase1980 » comme il est présenté ci-dessous. « Expbase1980 » est une copie de « eugene_1980_baseyear_cache » (voir Tableau 1). Elle contient donc notamment les tables name_model_coefficients et name_model_specifications dans lesquelles on trouve les paramètres à estimer. Par ailleurs, les données de sortie seront

également stockées dans «Expbase1980». Les données d'entrée, et plus particulièrement les paramètres estimés, seront écrasés et remplacés par les coefficients après estimation. Concrètement, seuls les paramètres des tables `name_model_coefficients` et `name_model_specifications` (`land_price_model_coefficients` et `land_price_model_specifications` dans notre cas) seront modifiés. Les autres données utilisées pour effectuer l'estimation seront donc recopiées sans modification. Ceci nous permet de limiter le nombre de bases de données à prendre en compte. Nous verrons que ceci n'est pas possible dans la phase de simulation où l'on garde les résultats de chaque année simulée afin de procéder ensuite à l'analyse.

```
'input_configuration':DatabaseConfiguration(
    host_name = os.environ.get('MYSQLHOSTNAME','localhost'),
    user_name = os.environ.get('MYSQLUSERNAME',''),
    password = os.environ.get('MYSQLPASSWORD',''),
    database_name = 'expbase1980',
),
'output_configuration':DatabaseConfiguration(
    host_name = os.environ.get('MYSQLHOSTNAME','localhost'),
    user_name = os.environ.get('MYSQLUSERNAME',''),
    password = os.environ.get('MYSQLPASSWORD',''),
    database_name = 'expbase1980',
),
```

3. ***estimation_runner.py*** (situé dans `C:\Python24\Lib\site-packages\urbansim\estimation`)

appelle les scripts des modèles spécifiés par `run_estimation.py` et établit le lien avec ceux comportant leurs spécifications. Ce lien est décrit par l'extrait ci-dessous. Chaque modèle est nommé de trois manières différentes: `initialedumodèle`, `INITIALEDUMODELE` et `name_model` référant à autant de scripts différents. La première renvoie au script `run_estimation.py` déjà présenté, la deuxième à celui précisant les spécifications (`LPM_specifications.py`) et le troisième à celui qui définit le modèle (`land_price_model.py`). Chaque modèle possède donc trois noms le caractérisant.

```
models = {
    "hlcm": ("HLCM", "household_location_choice_model"),
    "elcm-industrial": ("ELCM", "employment_location_choice_model",
"industrial", False),
    "elcm-commercial": ("ELCM", "employment_location_choice_model",
"commercial", False),
    "elcm-home_based": ("ELCM", "employment_location_choice_model",
"home_based", True),
    "dplcm-industrial": ("DPLCM", "development_location_choice_model",
"industrial"),
    "dplcm-commercial": ("DPLCM", "development_location_choice_model",
"commercial"),
    "dplcm-residential": ("DPLCM", "development_location_choice_model",
"residential"),
    "nlpm": ("NLPM", "new_land_price_model"),
    "lpm": ("LPM", "land_price_model"),
    "rlsm": ("RLSM", "residential_land_share_model")
}
```

4. ***land_price_model.py*** (situé dans C:\MyNew\projects\OPUS\opus-4.0\eugene\estimation ou EugTest\eugene\models dans Eclipse)

appelle le modèle à mettre en œuvre (regression_model.py), précise la variable dépendante de la régression (partie gauche de la régression, ici ln_total_land_value) qui est utilisée lors de l'estimation pour calculer les coefficients.

Il est à noter également que les noms identifiant le modèle sont rappelés au départ (lpm et land_price_model). Nous reviendrons plus en détail sur ce modèle dans la partie simulation.

5. ***LPM_specifications.py*** (situé dans C:\MyNew\projects\OPUS\opus-4.0\eugene\estimation ou EugTest\eugene\models dans Eclipse)

spécifie l'ensemble des variables explicatives à prendre en compte pour estimer les coefficients. Ces variables sont tirées de la table « gridcells » présentée précédemment. Elles sont spécifiées par leur nom précédé de l'adresse du dossier auquel elles appartiennent, ainsi que par leur abréviation.

```
specification={
-2:
[
('constant', 'constant'),
('urbansim.gridcell.percent_developed_within_walking_distance',
'PCTDEVWAL'),
('urbansim.gridcell.is_near_highway', 'HWY'),
('urbansim.gridcell.ln_distance_to_highway', 'BLDHW'),
('urbansim.gridcell.is_in_floodplain', 'BFLOOD'),
('opus_core.func.ln(urbansim.gridcell.total_improvement_value)',
'LN_IMPVAL'),
('eugene.gridcell.travel_time_hbw_am_drive_alone_to_220', 'CBDTRVL'),
('urbansim.gridcell.is_in_wetland', 'BWET'),
('urbansim.gridcell.ln_residential_units', 'BLDU'),
('urbansim.gridcell.ln_residential_units_within_walking_distance',
'BLDUW'),

('urbansim.gridcell.percent_high_income_households_within_walking_distance',
'BPHIW'),

],
}
```

Finalement, après avoir contrôlé ces cinq scripts et s'être assuré d'avoir fourni toutes les informations ou autres spécifications, l'estimation est lancée par l'exécution du script run_simulation.py. Un contrôle simple du fonctionnement de l'estimation et de l'enchaînement de

ces scripts est de supprimer des variables dans LPM_specifications.py puis de s'assurer à l'aide de MySQL que les changements ont été pris en compte dans les tables land_pice_model_specification et land_price_model_coefficients qui vont de paire.

3.2.3 La simulation

Après s'être assuré du bon fonctionnement de l'estimation, le cœur d'Urbansim réside sur les simulations à mettre en place. De la même manière que nous avons abordé la présentation de l'estimation, nous allons décrire cette phase de simulation en s'appuyant sur les différents scripts importants. Nous soulignerons ainsi les enchaînements des étapes, la localisation des informations à préciser. Cette démonstration proposera également une attention particulière à l'exemple du Land Price Model déjà évoqué dans le paragraphe précédent.

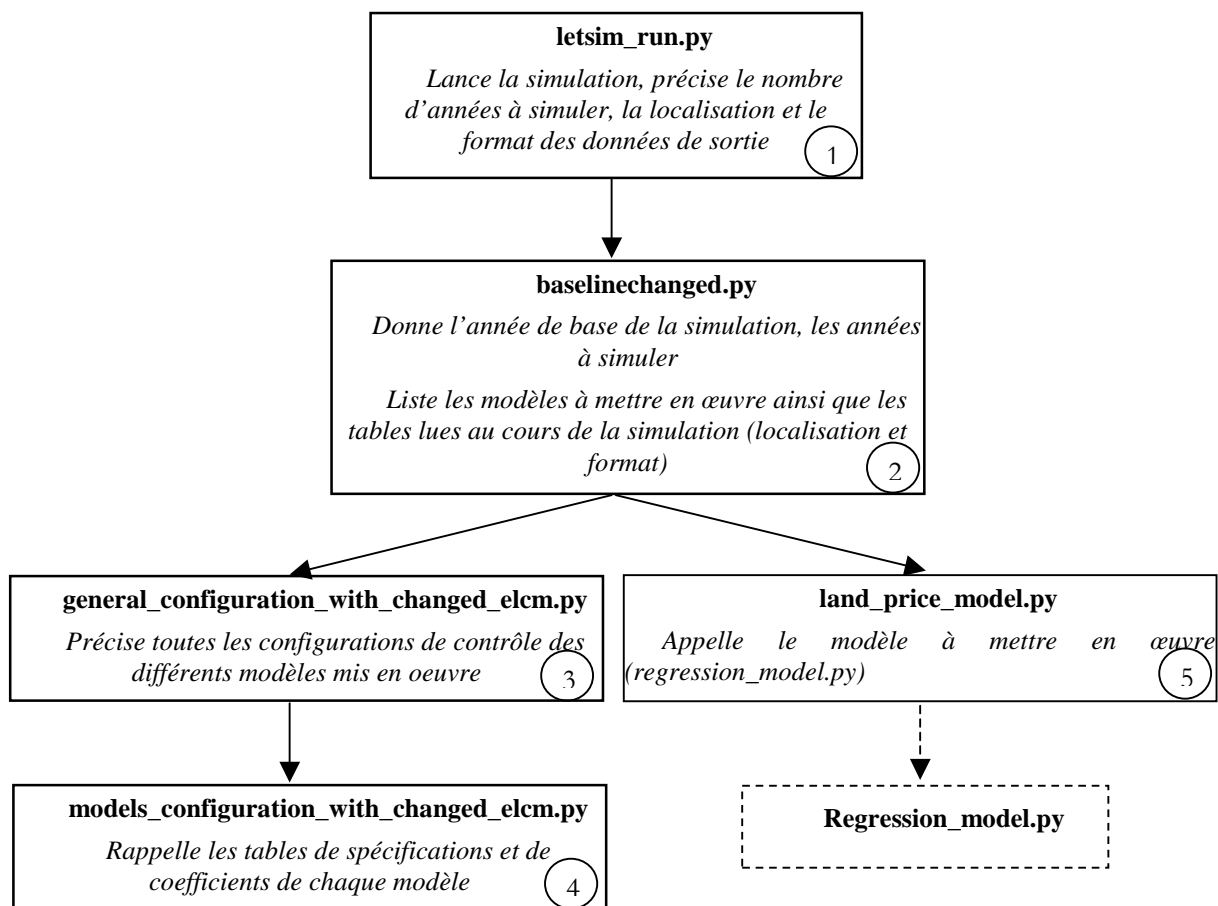


Figure 3-4 : Description de l'architecture et des scripts importants de la simulation

Comme le montre la figure ci-dessus, la simulation repose sur quatre principaux scripts, le cinquième étant celui de l'exemple du Land Price Model. Nous allons décrire plus en précision ces

quatre scripts afin de souligner les différentes informations qu'ils apportent ainsi que les connexions qu'ils établissent avec d'autres scripts ou avec les bases de données.

Il est important de rappeler au préalable que dans cette partie, des modifications ont été apportées à deux scripts originaux car des problèmes subsistaient lors des simulations effectuées avec ces scripts. Ces deux scripts sont `letsim_run.py` et `baselinechanged.py` qui à l'origine sont `run_simulation_on_baseyear_cache.py` (C:\Python24\Lib\site-packages\eugene\tools) et `baseline.py` (C:\Python24\Lib\site-packages\eugene\configs).

1. **`letsim_run.py`** (situé dans C:\MyNew\projects\OPUS\opus-4.0\eugene\tools ou EugTest\eugene\tools dans Eclipse)

est le script a exécuté pour lancer une simulation. Il fournit le nombre d'années à simuler, la localisation des données de sortie à stocker ainsi que le format de lecture des différentes bases de données. Suivant notre volonté de travailler avec des données au format MySQL, il est cependant nécessaire de le préciser dans le codage en choisissant de coder « True » comme le montre l'extrait ci-dessous. Par défaut, les données sont lues au format de base (.li4, .lf4 et .lf8 vus plus haut) donné dans le paquet Eugene, le code est alors « False ».

Trois lignes extraites du code de `letsim_run.py` :

```
output_directory= 'C:\output_test'  
years_to_run = 2  
  
[...]  
  
config['creating_baseyear_cache_configuration']['cache_from_mysql'] = True
```

2. **`baselinechanged.py`** (situé dans C:\MyNew\projects\OPUS\opus-4.0\eugene\configs ou EugTest\eugene\configs dans Eclipse)

est, comme son nom l'indique, le script de base de la simulation. `letsim_run.py` est le déclencheur de la simulation et `baselinechange.py` en est le cœur. Il contient les informations essentielles à la réalisation de la simulation. `Baselinechanged.py` informe de l'année de base sur laquelle est lancée la simulation ainsi que toutes les années à simuler.

```
config_changes = {  
    'description': 'Eugene baseline',  
    'base_year': 1980,  
    'years': (1981, 1982),
```

Le choix des modèles ainsi que des tables de données à mettre en œuvre au cours de la simulation sont également précisées. Ce choix se fait de la même manière que dans le cas de `run_estimation.py` vu plus haut. C'est à dire que les modèles qui ne sont pas appelés au cours de la simulation sont précédés d'un # sur la ligne de codage (voir

extrait du script ci-dessous). Ceci est une règle de codage du logiciel Python.

```
'models': [  
    #'prescheduled_events',  
    #'events_coordinator',  
    #'residential_land_share_model',  
    'land_price_model',  
    #'development_project_transition_model',  
    #'residential_development_project_location_choice_model',  
    #'commercial_development_project_location_choice_model',  
    #'industrial_development_project_location_choice_model',  
    #'development_event_transition_model',  
    #'events_coordinator',  
    #'residential_land_share_model',  
    #'household_transition_model',  
    #'employment_transition_model',
```

Le choix des tables présentées dans le Tableau 1 s'établit de la même façon. Elles sont énumérées et sont appelées si elles ne sont pas précédées d'un #. Par ailleurs, leur localisation et leur format est également rappelé dans ce script. Nous pouvons voir ici que nous utilisons la base de données obtenue au cours de l'estimation comme base de données d'entrée (« expbase1980 »). Elle comporte donc bien les paramètres estimés des modèles. Il est nécessaire de s'assurer que le format de lecture MySQL est bien activé (« True »).

```
'in_storage_type': 'mysql',  
    'input_configuration': DatabaseConfiguration(  
        host_name      = os.environ.get('MYSQLHOSTNAME', 'localhost'),  
        user_name      = os.environ.get('MYSQLUSERNAME', ''),  
        password       = os.environ.get('MYSQLPASSWORD', ''),  
        database_name  = 'expbase1980',  
  
    [...]  
  
    'cache_from_mysql': True,
```

3. *general_configuration_with_changed_elcm.py* (situé dans C:\Python24\Lib\site-packages\urbansim\configs)

rappelle l'ordre dans lequel sont exécutés. Cet ordre est le même que celui présenté dans *baselinechanged.py*. Il précise par ailleurs les configurations de contrôle de tous les modèles. On peut voir ainsi que chaque configuration de contrôle fait référence aux tables de spécification et de coefficients présentées précédemment.

```
_controller_configuration["land_price_model"] = Configuration({
    "import": {"urbansim.models.land_price_model":
               "LandPriceModel",
              },
    "init": {
        "name": "LandPriceModel"},
    "prepare_for_run": {
        "name": "prepare_for_run",
        "arguments": {"specification_storage":
_models_configuration["specification_storage"],
                    "specification_table": "%s" %
_models_configuration["land_price_model"]['specification_table'],
                    "coefficients_storage":
_models_configuration["coefficients_storage"],
                    "coefficients_table": "%s" %
_models_configuration["land_price_model"]['coefficients_table'],
                    },
        "output": "(specification, coefficients)"
    },
    "run": {
        "arguments": {"n_simulated_years": "year-resources['base_year']",
                    "specification": "specification",
                    "coefficients": "coefficients",
                    "dataset": "gridcell",
                    "data_objects": "datasets" ,
                    "chunk_specification": "'{ 'nchunks':1}",
                    "debuglevel": config['debuglevel']}
```

4. *models_configuration_with_changed_elcm.py* situé dans (C:\Python24\Lib\site-packages\urbansim\configs)

précise pour chaque modèle le nom de ses tables de spécification et de coefficients comme le montre l'extrait ci-dessous. Ce script est lié au script présenté juste précédemment.

```
'land_price_model':{
    "specification_table": 'land_price_model_specification',
    "coefficients_table": 'land_price_model_coefficients',
```

5. *land_price_model.py* (situé dans C:\MyNew\projects\OPUS\opus-4.0\eugene\estimation ou EugTest\eugene\models dans Eclipse)

comme tous les autres modèles, il est appelé seulement s'il n'est pas précédé d'un # dans le code du script *baselinechanged.py*. Il permet de calculer les valeurs de *residential_land_value* et *non_residential_value*. Celles de *residential_land_value* sont calculées par le modèle de régression hédonique (*regression_model.py*) à partir des paramètres calculés lors de l'estimation ainsi qu'à partir des données d'entrée de la table « *gridcells* » (constant, percent_developed_within_walking_distance, is_near_highway, ln_distance_to_highway, etc. voir extrait du script *LPM_specification.py*) spécifiées par les tables *land_price_model_coefficients* et *land_price_model_specification*. Ainsi, une fois le modèle de régression exécuté, le script *land_price_model.py* permet de calculer les valeurs de *non_residential_value* comme il est présenté dans l'extrait du code. Ces valeurs sont alors stockées dans la table « *gridcells* » de la base de données de sortie de la simulation.

```
def run(self, specification, coefficients, dataset, index=None,
chunk_specification=None,
        data_objects=None, run_config=None, debuglevel=0):
    """ For info on the arguments see RegressionModel.
    """
    if self.filter <> None:
        res = Resources(data_objects)
        res.merge({"debug":debuglevel})
        index = dataset.get_filtered_index(self.filter, threshold=0, index=index,
resources=res)
        ln_total_land_value = RegressionModel.run(self, specification, coefficients,
dataset, index, chunk_specification, data_objects,
                                                run_config, debuglevel)
        if (ln_total_land_value == None) or (ln_total_land_value.size() <=0):
            return ln_total_land_value
        if index == None:
            index = arange(dataset.size())
        total_land_value = exp(ln_total_land_value)
        residential_land_value = total_land_value *
dataset.get_attribute_by_index("fraction_residential_land", index)
        nonresidential_land_value = total_land_value - residential_land_value
        dataset.set_values_of_one_attribute("residential_land_value",
residential_land_value, index)
        dataset.set_values_of_one_attribute("nonresidential_land_value",
nonresidential_land_value,
                                        index)

        self.post_check(dataset)
        return index
```


3.2.4 Modifier ou créer un modèle

Après avoir étudié l'organisation et l'architecture d'Urbansim, le dernier point que nous allons aborder maintenant se place dans l'optique des futurs travaux sur les bases de données et l'aire urbaine lyonnaises. En procédant à ces descriptions dans les paragraphes précédents, nous avons non seulement présenté le fonctionnement général du modèle Urbansim, mais aussi relevé les éléments et scripts contenant les informations importantes à prendre en compte en terme de modélisation et de codage. Dans les arguments qui ont joué en faveur du choix d'Urbansim dans le cadre du projet SIMBAD on trouvait notamment sa souplesse en terme de codage et de modélisation. Bien que les premières analyses furent difficiles à mettre en place du fait de la complexité ainsi que du nombre important de scripts et dossiers du modèle, nous avons pu éclaircir cette organisation et retirer les éléments faisant son ossature.

Nous avons donc par la suite cherché à modifier un modèle et ses caractéristiques : son nom, les données sur lesquelles il s'appuie, les paramètres choisis, les indicateurs à étudier. Comme le soulignent les analyses de l'estimation et de la simulation, une attention particulière a été portée au Land Price Model.

Notre volonté a été tout d'abord de créer un modèle appelé New Land Price Model basé sur le même modèle de régression linéaire que Land Price Model. Un changement important opéré a été celui des variables de spécification prises en compte. Voyons dès lors quels sont les scripts affectés par ces modifications et à quel niveau elles doivent être faites.

3.2.4.1 *Au niveau de l'estimation*

Concernant l'estimation, tous les scripts présentés connaissent des modifications. Tout d'abord, il est indispensable de nommer le nouveau modèle suivant les trois appellations qui le caractérisent dans les différents scripts, de la même manière que cela a été montré pour le Land Price Model.

- New Land Price Model est son nom (pour Land Price Model)
- `new_land_price_model` (pour `land_price_model`)
- `nlpm` est son abréviation (pour `lpm`)
- LPM pour `lpm`

Souhaitant construire un nouveau modèle basé sur le même modèle de régression que celui de Land Price Model, nous avons créé un fichier `new_land_price_model.py` comprenant le même code que `land_price_model.py` en changeant les différentes appellations faisant référence au Land Price Model par celles correspondant au New Land Price Model comme ci-dessus.

De la même manière il est nécessaire de créer un script `NLPM_specification.py` dans lequel nous avons changé les spécifications à retenir. Ceci a permis de constater que ce script entrainait bien en vigueur dans l'enchaînement des scripts comme le montre les figures suivantes tirées de l'interface MySQL. Ce tableau montre bien que le fait de spécifier de nouvelles variables dans le script `NLPM_specification.py` a bien été pris en compte. Ces modifications sont apportées d'une part en ajoutant le nom de variables suivant le mode décrit au niveau du script `LPM_specification.py` présenté plus haut, et d'autre part en supprimant d'autres variables soit en effaçant leur ligne de code ou soit en « masquant » cette ligne par un `#` apposé au devant.

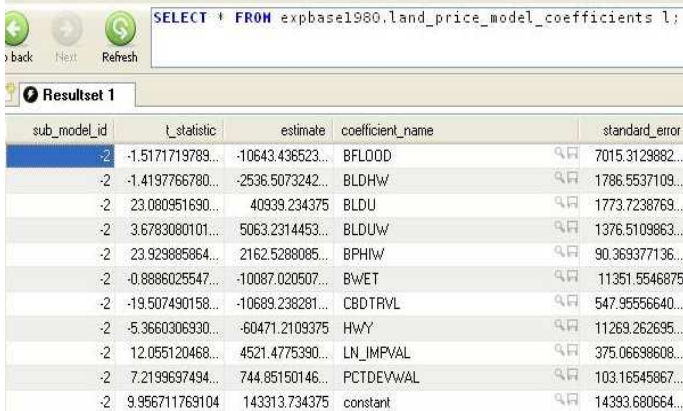
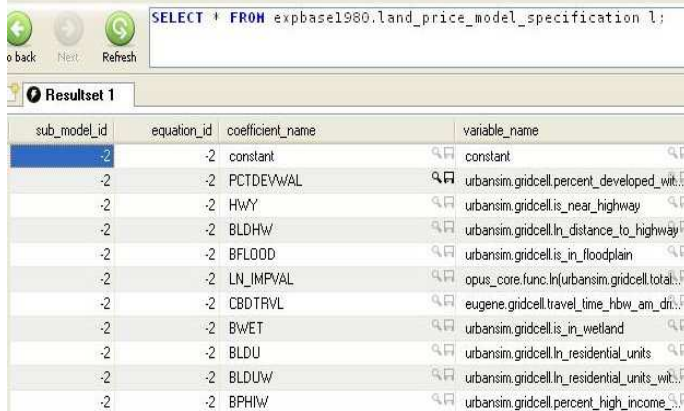
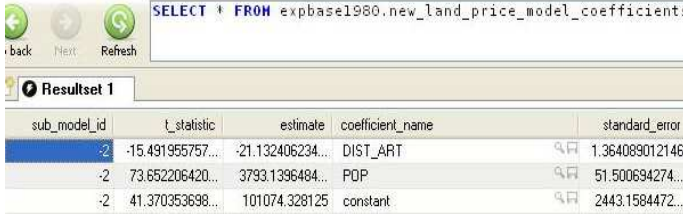

	Tables des coefficients	Tables des spécification																																																																																																												
Land Price Model	 <pre>SELECT * FROM expbase1980.land_price_model_coefficients l;</pre> <table border="1"> <thead> <tr> <th>sub_model_id</th> <th>t_statistic</th> <th>estimate</th> <th>coefficient_name</th> <th>standard_error</th> </tr> </thead> <tbody> <tr><td>-2</td><td>-1.5171719789...</td><td>-10643.436523...</td><td>BFLOOD</td><td>7015.3129882...</td></tr> <tr><td>-2</td><td>-1.4197766780...</td><td>-2536.5073242...</td><td>BLDHW</td><td>1786.5537109...</td></tr> <tr><td>-2</td><td>23.080951690...</td><td>40939.234375</td><td>BLDU</td><td>1773.7238769...</td></tr> <tr><td>-2</td><td>3.6783080101...</td><td>5063.2314453...</td><td>BLDUW</td><td>1376.5109863...</td></tr> <tr><td>-2</td><td>23.929885864...</td><td>2162.5288085...</td><td>BPHIW</td><td>90.369377136...</td></tr> <tr><td>-2</td><td>-0.8886025547...</td><td>-10087.020507...</td><td>BWET</td><td>11351.5546875</td></tr> <tr><td>-2</td><td>-19.507490158...</td><td>-10689.238281...</td><td>CBDTRVL</td><td>547.95556640...</td></tr> <tr><td>-2</td><td>-5.3660306930...</td><td>-60471.2109375</td><td>HWY</td><td>11269.262695...</td></tr> <tr><td>-2</td><td>12.055120468...</td><td>4521.4775390...</td><td>LN_IMPVAL</td><td>375.06698608...</td></tr> <tr><td>-2</td><td>7.2199697494...</td><td>744.85150146...</td><td>PCTDEVWAL</td><td>103.16545867...</td></tr> <tr><td>-2</td><td>9.956711769104</td><td>143313.734375</td><td>constant</td><td>14393.680664...</td></tr> </tbody> </table>	sub_model_id	t_statistic	estimate	coefficient_name	standard_error	-2	-1.5171719789...	-10643.436523...	BFLOOD	7015.3129882...	-2	-1.4197766780...	-2536.5073242...	BLDHW	1786.5537109...	-2	23.080951690...	40939.234375	BLDU	1773.7238769...	-2	3.6783080101...	5063.2314453...	BLDUW	1376.5109863...	-2	23.929885864...	2162.5288085...	BPHIW	90.369377136...	-2	-0.8886025547...	-10087.020507...	BWET	11351.5546875	-2	-19.507490158...	-10689.238281...	CBDTRVL	547.95556640...	-2	-5.3660306930...	-60471.2109375	HWY	11269.262695...	-2	12.055120468...	4521.4775390...	LN_IMPVAL	375.06698608...	-2	7.2199697494...	744.85150146...	PCTDEVWAL	103.16545867...	-2	9.956711769104	143313.734375	constant	14393.680664...	 <pre>SELECT * FROM expbase1980.land_price_model_specification l;</pre> <table border="1"> <thead> <tr> <th>sub_model_id</th> <th>equation_id</th> <th>coefficient_name</th> <th>variable_name</th> </tr> </thead> <tbody> <tr><td>-2</td><td>-2</td><td>constant</td><td>constant</td></tr> <tr><td>-2</td><td>-2</td><td>PCTDEVWAL</td><td>urbansim.gridcell.percent_developed_w...</td></tr> <tr><td>-2</td><td>-2</td><td>HWY</td><td>urbansim.gridcell.is_near_highway</td></tr> <tr><td>-2</td><td>-2</td><td>BLDHW</td><td>urbansim.gridcell.in_distance_to_highway</td></tr> <tr><td>-2</td><td>-2</td><td>BFLOOD</td><td>urbansim.gridcell.is_in_floodplain</td></tr> <tr><td>-2</td><td>-2</td><td>LN_IMPVAL</td><td>opus_core.func.in(urbansim.gridcell.total...</td></tr> <tr><td>-2</td><td>-2</td><td>CBDTRVL</td><td>eugene.gridcell.travel_time_hbw_am_dir...</td></tr> <tr><td>-2</td><td>-2</td><td>BWET</td><td>urbansim.gridcell.is_in_wetland</td></tr> <tr><td>-2</td><td>-2</td><td>BLDU</td><td>urbansim.gridcell.in_residential_units</td></tr> <tr><td>-2</td><td>-2</td><td>BLDUW</td><td>urbansim.gridcell.in_residential_units_w...</td></tr> <tr><td>-2</td><td>-2</td><td>BPHIW</td><td>urbansim.gridcell.percent_high_income...</td></tr> </tbody> </table>	sub_model_id	equation_id	coefficient_name	variable_name	-2	-2	constant	constant	-2	-2	PCTDEVWAL	urbansim.gridcell.percent_developed_w...	-2	-2	HWY	urbansim.gridcell.is_near_highway	-2	-2	BLDHW	urbansim.gridcell.in_distance_to_highway	-2	-2	BFLOOD	urbansim.gridcell.is_in_floodplain	-2	-2	LN_IMPVAL	opus_core.func.in(urbansim.gridcell.total...	-2	-2	CBDTRVL	eugene.gridcell.travel_time_hbw_am_dir...	-2	-2	BWET	urbansim.gridcell.is_in_wetland	-2	-2	BLDU	urbansim.gridcell.in_residential_units	-2	-2	BLDUW	urbansim.gridcell.in_residential_units_w...	-2	-2	BPHIW	urbansim.gridcell.percent_high_income...
	sub_model_id	t_statistic	estimate	coefficient_name	standard_error																																																																																																									
-2	-1.5171719789...	-10643.436523...	BFLOOD	7015.3129882...																																																																																																										
-2	-1.4197766780...	-2536.5073242...	BLDHW	1786.5537109...																																																																																																										
-2	23.080951690...	40939.234375	BLDU	1773.7238769...																																																																																																										
-2	3.6783080101...	5063.2314453...	BLDUW	1376.5109863...																																																																																																										
-2	23.929885864...	2162.5288085...	BPHIW	90.369377136...																																																																																																										
-2	-0.8886025547...	-10087.020507...	BWET	11351.5546875																																																																																																										
-2	-19.507490158...	-10689.238281...	CBDTRVL	547.95556640...																																																																																																										
-2	-5.3660306930...	-60471.2109375	HWY	11269.262695...																																																																																																										
-2	12.055120468...	4521.4775390...	LN_IMPVAL	375.06698608...																																																																																																										
-2	7.2199697494...	744.85150146...	PCTDEVWAL	103.16545867...																																																																																																										
-2	9.956711769104	143313.734375	constant	14393.680664...																																																																																																										
sub_model_id	equation_id	coefficient_name	variable_name																																																																																																											
-2	-2	constant	constant																																																																																																											
-2	-2	PCTDEVWAL	urbansim.gridcell.percent_developed_w...																																																																																																											
-2	-2	HWY	urbansim.gridcell.is_near_highway																																																																																																											
-2	-2	BLDHW	urbansim.gridcell.in_distance_to_highway																																																																																																											
-2	-2	BFLOOD	urbansim.gridcell.is_in_floodplain																																																																																																											
-2	-2	LN_IMPVAL	opus_core.func.in(urbansim.gridcell.total...																																																																																																											
-2	-2	CBDTRVL	eugene.gridcell.travel_time_hbw_am_dir...																																																																																																											
-2	-2	BWET	urbansim.gridcell.is_in_wetland																																																																																																											
-2	-2	BLDU	urbansim.gridcell.in_residential_units																																																																																																											
-2	-2	BLDUW	urbansim.gridcell.in_residential_units_w...																																																																																																											
-2	-2	BPHIW	urbansim.gridcell.percent_high_income...																																																																																																											
New Land Price Model	 <pre>SELECT * FROM expbase1980.new_land_price_model_coefficients</pre> <table border="1"> <thead> <tr> <th>sub_model_id</th> <th>t_statistic</th> <th>estimate</th> <th>coefficient_name</th> <th>standard_error</th> </tr> </thead> <tbody> <tr><td>-2</td><td>-15.491955757...</td><td>-21.132406234...</td><td>DIST_ART</td><td>1.364089012146</td></tr> <tr><td>-2</td><td>73.652206420...</td><td>3793.1396484...</td><td>POP</td><td>51.500694274...</td></tr> <tr><td>-2</td><td>41.370353698...</td><td>101074.328125</td><td>constant</td><td>2443.1584472...</td></tr> </tbody> </table>	sub_model_id	t_statistic	estimate	coefficient_name	standard_error	-2	-15.491955757...	-21.132406234...	DIST_ART	1.364089012146	-2	73.652206420...	3793.1396484...	POP	51.500694274...	-2	41.370353698...	101074.328125	constant	2443.1584472...	 <pre>SELECT * FROM expbase1980.new_land_price_model_specification r</pre> <table border="1"> <thead> <tr> <th>sub_model_id</th> <th>equation_id</th> <th>coefficient_name</th> <th>variable_name</th> </tr> </thead> <tbody> <tr><td>-2</td><td>-2</td><td>constant</td><td>constant</td></tr> <tr><td>-2</td><td>-2</td><td>POP</td><td>urbansim.gridcell.population</td></tr> <tr><td>-2</td><td>-2</td><td>DIST_ART</td><td>site-packages.distance_to_arterial</td></tr> </tbody> </table>	sub_model_id	equation_id	coefficient_name	variable_name	-2	-2	constant	constant	-2	-2	POP	urbansim.gridcell.population	-2	-2	DIST_ART	site-packages.distance_to_arterial																																																																								
sub_model_id	t_statistic	estimate	coefficient_name	standard_error																																																																																																										
-2	-15.491955757...	-21.132406234...	DIST_ART	1.364089012146																																																																																																										
-2	73.652206420...	3793.1396484...	POP	51.500694274...																																																																																																										
-2	41.370353698...	101074.328125	constant	2443.1584472...																																																																																																										
sub_model_id	equation_id	coefficient_name	variable_name																																																																																																											
-2	-2	constant	constant																																																																																																											
-2	-2	POP	urbansim.gridcell.population																																																																																																											
-2	-2	DIST_ART	site-packages.distance_to_arterial																																																																																																											

Tableau 5 : illustration depuis MySQL, de la prise en compte des modifications dans le choix des variables spécifiées depuis `NLPM_specification.py`.

3.2.4.2 Au niveau de la simulation

Pour ce qui est de la simulation, le script de base `letsim_run.py` n'est pas affecté par les éventuelles modifications. Cependant dans tous les autres scripts, il est nécessaire de rajouter les informations correspondant au New Land Price Model écrite avec la même forme que celles du Land Price Model, en opérant tout de même les changements d'appellations. Voici quelques exemples :

```
'models': [
    #'prescheduled_events',
    #'events_coordinator',
    #'residential_land_share_model',
    'new_land_price_model'
    #'land_price_model',
    #'development_project_transition_model',
    #'residential_development_project_location_choice_model',
    #'commercial_development_project_location_choice_model',
```

Figure 3-4 : Extrait du code du script de `baselinchanged.py` illustrant l'ajout du modèle NewLand Price Model.

```
'residential_land_share_model':{
    "specification_table": 'residential_land_share_model_specification',
    "coefficients_table": 'residential_land_share_model_coefficients',
},
'new_land_price_model':{
    "specification_table": 'new_land_price_model_specification',
    "coefficients_table": 'new_land_price_model_coefficients',
},
'household_location_choice_model':{
    "specification_table": 'household_location_choice_model_specification',
    "coefficients_table": 'household_location_choice_model_coefficients',
},
```

Figure 3-4 : Extrait du code du script de *models_configuration_with_changed_elcm.py* illustrant l'ajout du modèle NewLand Price Model.

3.3 Conclusion partielle

A travers cette description technique sous forme de guide d'utilisateur nous avons pu souligner et mettre en avant les éléments importants de modélisation d'UrbanSim :

- Prises en compte des données
- Leur affichage, leur extraction, leur représentation
- Repérage des scripts à exécuter et à modifier
- Compréhension de l'enchaînement de ces scripts

Cette prise en compte pourra servir de base de travail aux différentes personnes se joignant au projet collectif qu'est SIMBAD. Elle permettra également de mieux aborder le transfert et l'intégration des données lyonnaises au modèle qui devrait se faire dans les prochains mois.

4 CONCLUSION

A travers ce Travail de Fin d'Etudes s'insérant dans le projet SIMBAD développé au Laboratoire d'Economie des Transports depuis 2005, nous avons pu dresser une première présentation de la problématique de SIMBAD s'appuyant sur l'étude de la structure du modèle intégré des interactions transport et urbanisme UrbanSim.

Pour ce faire, une première partie a permis de présenter le projet global SIMBAD en présentant la mise en place de ce projet faisant suite au projet Télescopage qui n'apportait pas d'éléments de réponse à la problématique d'interaction de transport à urbanisme et qui ne s'inscrivait pas non plus dans une réflexion de mobilité durable.

Nous avons pu également voir dans cette première partie comment le projet SIMBAD s'articulait autour de la structure du modèle intégré UrbanSim dont nous avons fait la présentation des différents modules. Ainsi, cette description a pu souligner comment le lien entre transport et urbanisme dans un horizon à long terme basé sur la notion de mobilité durable se dégageait de cette organisation.

D'autre part, ce Travail de Fin d'Etude représente également une contribution technique et scientifique à un projet collectif porté par SIMBAD. En effet, la réalisation dans la deuxième partie du guide d'utilisateur relevant les éléments de modélisation à considérer afin de comprendre la structure globale du modèle et de bien aborder le passage aux bases de données lyonnaises sera un appui écrit disponible à toute personne se joignant au projet. Par cette description technique des points importants de l'architecture, il donne la possibilité de bien intégrer les enjeux en terme de modélisation et de traitement de données nécessaires à ce transfert du travail sur l'aire d'Eugen-Springfield à celui sur l'aire urbaine lyonnaise.

Enfin, ce travail est le point de départ de mon mémoire de recherche de Master. Après avoir acquis les connaissances techniques globales inhérentes au modèle UrbanSim, une attention particulière sera portée au module d'accessibilité présenté précédemment. Une réflexion sur cette notion sera menée parallèlement à des tests de simulation sur les données d'Eugene-Springfeld afin de mettre en avant les indicateurs et variables pouvant avoir une influence notable sur les valeurs d'accessibilité.

Cette réflexion aura également une portée dans le projet global SIMBAD. Les résultats de ces tests pourront ainsi permettre d'évaluer la manière avec laquelle doit être abordée cette notion à l'échelle lyonnaise et plus généralement à une échelle française voire européenne. Il existe en effet une importante différence entre les structures urbaines nord américaines et européennes. Cet aspect pourra donc influencer la prise en compte de cette notion ainsi que celle des indicateurs et variables à étudier dans le cas lyonnais.

5 BIBLIOGRAPHIE

DE PALMA *et al.* . SIMAURIF - Modèle dynamique de simulation de l'interaction Urbanisation - transports en région Île-de-France, Application à la tangentielle nord. Rapport final de 1ère année. Collection Recherche Transport, 148 p. . Banque CDAT TEMIS, côte : 15192. Ile de France : DRAST, 11/11/2004. Coordonnées disponible sur <http://temis.documentation.equipement.gouv.fr/temis/document.xsp?id=49918&qid=sdx_q0&n=1&q=> (consulté le 20.01.2007)

DEYMIER Ghislaine, NICOLAS Jean-Pierre. Modèles d'interaction entre transport et urbanisme : état de l'art et choix du modèle pour le projet SIMBAD. Rapport intermédiaire n°1. Rapport intermédiaire n°1 du projet SIMBAD Simuler les MoBilités pour Agglomération Durable [en ligne]. Pour le compte de la DRAST (Ministère de l'Équipement) et de l'ADEME dans le cadre du groupe 11 du PREDIT, Juillet 2005. Disponible sur <<http://halshs.archives-ouvertes.fr/docs/00/10/13/42/PDF/SimbadLuti.pdf>> (consulté le 20.01.2007).

HUNT *et al.*. Current Operational Urban Land-use–Transport, Modelling Frameworks: A Review. Transport Reviews, Vol. 25, No. 3, 329–376, Mai 2005.

MASSON. Les interactions entre système de transport et système de localisation en milieu urbain et leur modélisation. Thèse pour le doctorat de Sciences Economiques Mention Economie des Transports. Lyon : Université Lyon 2, Faculté de Sciences Economiques et de Gestion, décembre 2000, 549p.

NICOLAS *et al.* . Rapport n°4 : l'architecture du modèle au sein du projet SIMBAD. Pour le compte de la DRAST (Ministère de l'Équipement) et de l'ADEME dans le cadre du groupe 11 du PREDIT, Mars 2006. Disponible sur <<http://halshs.archives-ouvertes.fr/docs/00/10/13/99/PDF/ArchitectureSimbad.pdf>> (consulté le 27.01.2007).

TORRENS. How Land-Use/transportation Models Work. Paper20 of Centre for Advanced Spatial Analysis. Novembre 2000. 75p. ISSN : 1467-1298.

VERRY Damien, NICOLAS Jean-Pierre. Indicateurs de mobilité durable : de l'état de l'art à la définition des indicateurs dans le projet Simbad. Rapport intermédiaire n°2 du projet SIMBAD Simuler les MoBilités pour Agglomération Durable [en ligne]. Pour le compte de la DRAST (Ministère de l'Équipement) et de l'ADEME dans le cadre du groupe 11 du PREDIT, Juillet 2005. Disponible sur <<http://halshs.archives-ouvertes.fr/docs/00/10/13/52/PDF/SimbadIndicateurs.pdf>> (consulté le 27.01.2007).

WADDELL *et al.* . Land Use and Travel Demand Forecasting Models. Review of the Literature and Operational Models. Juin 2001. 156p.

WADDELL. UrbanSim : Modeling Urban Development for Land Use, Transportation, and Environmental Planning. Journal of the American planning association. (USA). Été 2002, Vol. 68, n° 3, pp. 297-314, tabl., graph., cartes, notes, bibliogr. ISSN : 0194-4363. Côte : CDU. Lieu : Eugène, Oregon. Résumé disponible sur <<http://216.239.59.104/search?q=cache:tQ1lZyQkQ4MJ:developpementdurable.documentation.e>

quipement.gouv.fr/developpementdurable/vuedocpdf%3Fid%3D246388%26print%3Dtrue+urbansim&hl=fr&ct=clnk&cd=6> (consulté le 07/05/07).

Site internet : www.urbansim.org, The UrbanSim Project, Juillet 2007

ANNEXES

ANNEXE A

Scripts for run simulation on Eugene in MyNew :

Script of letsim run.py (EugTest\eugene\tools, C:\MyNew\projects\OPUS\opus-4.0\eugene\tools) :

```
import os
import sys

from time import strftime, localtime
from opus_core.fork_process import ForkProcess
from opus_core.misc import create_import_for_camel_case_class

#
# Edit the following lines before running a simulation
#

cache_dir = 'C:\New_eugene_1980_baseyear_cache'
#cache_dir = 'C:\output_test\run_1.2007_07_17_12_18'
#cache_dir = 'C:\Program Files\MySQL\MySQL Server 5.0\data\baseyear1980'
output_directory= 'C:\output_test'
years_to_run = 2

#
# Do not edit below
#

output_dir = os.path.join(str(output_directory),
strftime('%Y_%m_%d_%H_%M', localtime()))
classpath = 'eugene.configs.baselinechanged'
exec(create_import_for_camel_case_class(classpath,
import_as='ImportedConfiguration'))

config = ImportedConfiguration()

base_year = config['creating_baseyear_cache_configuration']['base_year']

config['creating_baseyear_cache_configuration']['cache_from_mysql'] = True
```

```
config['creating_baseyear_cache_configuration']['baseyear_cache']['directory_to_cache'] = cache_dir
config['creating_baseyear_cache_configuration']['cache_directory'] = output_dir
config['years'] = (base_year+1, base_year+years_to_run)

if not len(range(config['years'][0], config['years'][1]+1)) > 0:
    print 'Nothing to simulate!'
    sys.exit(1)

ForkProcess().fork_new_process('opus_core.tools.start_run', resources=config)
```

Script of baselinechanged.py (EugTest\eugene\configs, C:\MyNew\projects\OPUS\opus-4.0\eugene\configs):

```
import os

from opus_core.configurations.database_configuration import DatabaseConfiguration
from urbansim.configs.base_configuration import AbstractUrbansimConfiguration
from urbansim.configs.general_configuration_with_changed_elcm import
GeneralConfiguration

class Baselinechanged(GeneralConfiguration):
    """Eugene's baseline configuration.
    """
    def __init__(self):
        config = AbstractUrbansimConfiguration()

        config_changes = {
            'description': 'Eugene baseline',
            'base_year': 1980,
            'years': (1981, 1982),
            'models': [
                'prescheduled_events',
                'events_coordinator',
                'residential_land_share_model',
                'land_price_model',
                'development_project_transition_model',
                'residential_development_project_location_choice_model',
                'commercial_development_project_location_choice_model',
                'industrial_development_project_location_choice_model',
                'development_event_transition_model',
                'events_coordinator',
                'residential_land_share_model',
                'household_transition_model',
                'employment_transition_model',
                'household_relocation_model',
                'household_location_choice_model',
                'employment_relocation_model',
                {
                    'employment_location_choice_model': {
                        'group_members':
'_all_'}}},
                'distribute_unplaced_jobs_model',
            ],
```

```

'in_storage_type': 'mysql',


```

```
'commercial_employment_location_choice_model_coefficients',
'commercial_employment_location_choice_model_specification',
'home_based_employment_location_choice_model_specification',
'home_based_employment_location_choice_model_coefficients',
'industrial_employment_location_choice_model_coefficients',
'industrial_employment_location_choice_model_specification',
'industrial_development_location_choice_model_coefficients',
'industrial_development_location_choice_model_specification',
'residential_development_location_choice_model_coefficients',
'residential_development_location_choice_model_specification',
#'fazes',
'urbansim_constants',
'household_location_choice_model_coefficients',
'household_location_choice_model_specification',
'land_price_model_coefficients',
'land_price_model_specification',
'residential_land_share_model_coefficients',
'residential_land_share_model_specification',
#'plan_type_group_definitions',
#'plan_type_groups',
#'large_areas',
'household_characteristics_for_ht',
'development_types',
'development_type_group_definitions',
'development_constraints',
'annual_relocation_rates_for_households',
'annual_relocation_rates_for_jobs',
'base_year',
'cities',
#'development_constraint_events',
#'development_events',
'development_type_groups',
'employment_adhoc_sector_group_definitions',
'employment_adhoc_sector_groups',
'employment_events',
'employment_sectors',
#'land_use_events',
'plan_types',
'race_names',
'target_vacancies'
#'jobs_for_estimation',
```

```
        #'households_for_estimation',
        #'development_events_exogenous',
        ],
        'tables_to_cache_nchunks': {'gridcells': 1},
        'tables_to_copy_to_previous_years': {
            #'development_type_group_definitions': 1975,
            #'development_type_groups': 1975,
            #'development_types': 1975
        },
    },
}
config.merge(config_changes)
self.merge(config)
```


ANNEXE B

Scripts for run estimation on Eugene in MyNew :

Script of run estimation.py (C:\MyNew\projects\OPUS\opus-4.0\eugene\estimation) :

```
from eugene.estimation.my_estimation_config import my_configuration
#from urbansim.estimation.estimation_runner import EstimationRunner
from estimation_runner import EstimationRunner
from opus_core.resources import Resources
from opus_core.store.mysql_database_server import MysqlDatabaseServer
from opus_core.storage_factory import StorageFactory

#model = 'hlcm'
#model = 'elcm-industrial'
#model = 'elcm-commercial'
#model = 'elcm-home_based'
#model = 'dplcm-industrial'
#model = 'dplcm-commercial'
#model = 'dplcm-residential'
model = 'lpm'
#model = 'rlsm'

estimator = EstimationRunner(model, specification_from_module=True,
package="eugene",
configuration=my_configuration,
save_estimation_results=True)

db_server = MysqlDatabaseServer(my_configuration['output_configuration'])
output_db = db_server.get_database(my_configuration['output_configuration'].database_name)
my_out_storage = StorageFactory().get_storage("mysql_storage",
resources=Resources({"storage_location":output_db}))
#dir = 'C:\urbansim_cache'
#resLoc = Resources({'storage_location':dir})
#my_out_storage = StorageFactory().get_storage("flt_storage", resources=resLoc)
estimator.estimate(out_storage=my_out_storage)
output_db.DoQuery('create table simbad_tmp (coll char (10))')
output_db.DoQuery('drop table if exists simbad_tmp')
output_db.close()
```



```
#estimator.save_results(out_storage=my_configuration['output_configuration'],
model_name=model)
#estimator.reestimate("HLCM_specification")
#estimator.reestimate("ELCM_specification", type="commercial")
```

Script of my estimation config.py (C:\MyNew\projects\OPUS\opus-4.0\eugene\estimation):

```
import os

from opus_core.configurations.database_configuration import DatabaseConfiguration

my_configuration = {
    'creating_baseyear_cache_configuration':{
        #'cache_directory':"/urbansim_cache/eugene", # change or leave out
        #'cache_directory':'C:\eugene_1980_baseyear_cache',
        'cache_directory':'C:\Old_eugene_1980_baseyear_cache',
        #'cache_directory':'C:\My325',
        'cache_from_mysql':True,
        'baseyear_cache':{
            #'directory_to_cache':"/urbansim_cache/eugene",
            'directory_to_cache':"C:/urbansim_cache",
            #'years':range(1996,2001)
        },
        'tables_to_cache_nchunks':{
            'gridcells':1
        },
        'unroll_gridcells':True,
    },
    'input_configuration':DatabaseConfiguration(
        host_name = os.environ.get('MYSQLHOSTNAME','localhost'),
        user_name = os.environ.get('MYSQLUSERNAME',''),
        password = os.environ.get('MYSQLPASSWORD',''),
        #database_name = 'Eugene_1980_baseyear_opus',
        database_name = 'expbase1980',
        #database_name = 'my325',
    ),
    'output_configuration':DatabaseConfiguration(
        host_name = os.environ.get('MYSQLHOSTNAME','localhost'),
        user_name = os.environ.get('MYSQLUSERNAME',''),
        password = os.environ.get('MYSQLPASSWORD',''),
        #database_name = 'Eugene_1980_baseyear_opus_estimation',
        database_name = 'expbase1980',
        #database_name = 'my325',
    ),
}
```

```
'datasets_to_cache_after_each_model': [],  
'low_memory_mode': False,  
'base_year': 1980,  
'years': (1980, 1980),  
'seed': (10, 10)  
}
```