Simulating Urban Growth Using Cellular Automata

A case study in Zhongshan city, China

Sun, Zhongwei
Feb 26th, 2003

Simulating Urban Growth Using Cellular Automata

A case study in Zhongshan city, China

by

Sun Zhongwei

Thesis submitted to the International Institute for Geo-information Science and Earth Observation in partial fulfilment of the requirements for the degree of Master of Science in urban planning and land administration.

Degree Assessment Board

Chairman:	Prof.Dr. W.H. van den Toorn
First Supervisor:	Dr.Ir. L.G.J. Boerboom
Second Supervisor:	Dr.Ir. J. Turkstra
Ext. Examiner:	Dr. F. Toppen

INTERNATIONAL INSTITUTE FOR GEO-INFORMATION SCIENCE AND EARTH OBSERVATION
ENSCHEDE, THE NETHERLANDS

Disclaimer

This document describes work undertaken as part of a programme of study at the International Institute for Geo-information Science and Earth Observation. All views and opinions expressed therein remain the sole responsibility of the author, and do not necessarily represent those of the institute.

Abstract

Urban growth has been the focus of urban researchers for a long time. Due to the complexity of the urban growth process, it is very difficult to model urban growth using traditional urban models. Recently, urban researchers are keen on a new kind of urban modelling method, Cellular automata.

Cellular Automata (CA) model was under intensive investigating since 1970s and was applied in many disciplines due to its ability of representing complex system dynamics. In urban research field, researchers set up many CA models to simulate the urban phenomena such as sprawl process, pedestrian dynamics, emergency evacuation, land use change trends, population migration, etc.

In this research, a CA model, ZSSIM, for simulating the urban growth process in a case study city, Zhongshan, China, has been developed and implemented within Ascape multi-agent developing environment using JAVA language. This model uses a multi criteria evaluation (MCE) transition rule and a root agent, which is a kind of central control mechanism, together to control the non-urban to urban transition process.

In this research, the urban growth process was assumed to be a one way transition, namely no anti urbanization process is considered. Five factors were considered to have impacts on urban growth: neighbourhood effect, road network, urban centres, terrain constraint, and policy. The effect of these factors was evaluated by the MCE transition rule and result in a developing potential value, which indicates the transition potential. The root agent ranks all the potential values into descend order and get a threshold for cell transition according to the annually land consumption. Finally the cells are transited according to their ranking positions compared with the threshold. By manipulating the weight system for different factors, the model is able to generate interpretable scenarios.

To simulate the trend break in the study city in 1992, which is cause by policy change and economic boom, a central control mechanism was introuce into this CA model. The root agent was defined to take care of the model data and parameters. The simulation is separated into two periods, from 1980 to 1992 and from 1992 to 1998. In 1992, the root agent will change the model control parameters and road network layer according to the second period's situation.

This research also calibrated the model by manually approach and assessed the best-fit result in a cell-to-cell match approach.

The conclusions on this research are: CA model using MCE controlled transition rule is a useful tool for modeling urban growth, ZSSIM can be a basis for further development of a tool for studying and predicting urban growth process, modelling trend break, which is usually caused by top down process, needs to introduce central control mechanism into CA model, manually calibration and visual evaluation are acceptable and feasible approaches in CA urban grow models.

Acknowledgement

When I am writing this, I think I can happily say that I have done my research finally. Now it is time to express my deepest gratitude to all the people who have supported and helped me in the completion of this research work.

First of all, I would like to express my gratitude to my supervisor, Mr. Luc Boerboom. Thanks for the intensive discussions and regular comments lasting through the whole research period. Without these, it is not possible for me to undertake and finish this research. I must say, I have learned a lot from you and that will eventually contributes to my future professional career.

Mr. Jan Turkstra, my second supervisor, I would like to say that your suggestions and encouragement helped me through this research. Special thanks to you for your suggestion at the early stage of the research which finally ensure me with the completion.

Many thanks to Mr Mark Brussel, Mr. Frans van den Bosch for they offered me a lot of support and to all staff of the Division of Urban Infrastructure for their support and help.

Special thanks go to Mr. Cheng jianquan, from the very beginning of this research till the very end, whenever my research was in trouble, your suggestions and knowledge could always help me through it.

To Mr. Li Sheng, my friend and field work contact person, I would like to thank you for your enormous help in the field and even after. Without your help, I could not carry out the fieldwork activities in the very short time available.

My thanks go to those people who gave me a lot of advices and opinions about my research, they are Mr.Zhan Qingming and Mr. Huang Zhengdong, thank you very much for you gave me helpful suggestions.

I express my gratitude to my organization, The Provincial surveying and mapping bureau of Heilongjiang, China. My organization and ITC together gave me the opportunity to undertake my research here.

I would like to express my gratitude to all the classmates of UPLA 2001, for John, Li, Weifeng, Yuayan, Katia, Chacra, Talat, Mercy, Nzau, and Busa for their encouragement, friendship and all the pleasure that we have shared. I also appreciate all my friends who gave me support and friendship that encouraged me through the research.

Special thank to my friends Li, Xiang and Zheng, we have had happy time together and you gave me enormous support and help in the past 18 months.

Last but not least, my thanks go to my family, thanks my mother and sisters for their always encouragement, understanding and support to me

Sun, Zhongwei
February 2003

Table of Content

List of Tables

List of Figures

1. Introduction

1.1. Background

Urban sprawl, rapid urban population growth, infrastructure services lagging behind the development of the urban area, city development not following master plans, all these phenomena lead to such questions as: what is going on during the urban development? What are the essential factors that affect the urban development? How does a real city work? How these factors interact with each other? These questions arise the fundamental question: How to understand the urban development process?

Understanding the urban growth process is very important in the urban planning, urban management, and policymaking process. Through a better understanding of the mechanism of urban development process, the urban planner and manager could be able to reduce the risk of making wrong or bad decisions and enhance the effectiveness of new plans. Many efforts have been made to answer these questions. Researches set up many models and invent many assistant tools to get a better understanding of the urban expansion phenomena. With the help of models, which are the simplified or abstracted representation of the real city, researchers and urban planners are trying to study the urban development process thoroughly.

The city in itself is a complex system. They are open, complex, far from equilibrium, and self-organising systems. There are two kinds of process simultaneously existing in a city together form up urban growth: top-down processes and self-organising processes. Top-down process denotes to the central control of the government and policies while self-organising process denotes the development caused by spontaneous decision-making process of private businesses and individual citizens in the city.

To understand a system, researchers usually build a model, which is an abstract of the real world, to simulate the dynamics of the target system. Traditionally, simulation model techniques are based on mathematical and stochastic models, generally based on differential equations, which relate various parameters and describe the dynamics of the systems. But when dealing with human activities, this type of aggregation models encounters problems(Grueau et al., 1999). As Batty mentioned in 1998, the issues linked with urban change are difficult to predict specifically, implying that our knowledge of the detailed dynamics of such change is at best rudimentary (Batty, 1998). To overcome these problems and utilize the rapid increasing computation capacity, urban modellers are looking for the new methods to model the urban phenomena.

Recently, researchers are keen on a new kind of urban simulation model: individual-based simulation models. Among them, the Cellular Automata (CA) model has been widely applied to the simulation of urban growth and forms. It is described as the new trend of urban modelling and is believed to be the most promising one that can be used to simulate the dynamics of the urban developing process.

1.2. Research problem

Many researches such as Batty, Couclelis, etc, show that CA has a potential usability to be applied in modelling change of the urban form. There are two bifurcations, one is to build artificial cities in pure academic research purpose and another type tries to mimic the real city in the operational level. Still there are many difficulties in simulating the dynamics of a real city. Limited by data, local knowledge and maybe the model itself, simulating the real city developing process in exactly the same way the city acts is not realistic. However, it is useful of building a CA model for a real city, to answer some "What-if" questions, namely, to build a scenario generating tool which can examine the policies and to build a "lab" to study the influence of some important issues in urban developments such as transportation, population immigration, etc.

In this research, one CA model prototype will be developed and implemented to study the urban growth and urban form changing process in Zhongshan city, China. Master plans will be translated into the model structure to regulate the model developing as it was in the real world guiding the urban development. After implementation of the CA model, it will generate some urban form scenarios and thus being assessed to check its usability of being used as an assistant tool for urban planning. Since Zhongshan city's development varied very much in two periods, from the start of 1980s to the start of 1990s and from the start of 1990s to the end of 1990s, in spatial pattern and temporal aspect, to present the two different periods of development, the CA model will use different parameter setting for the two periods.

1.3. Research objectives

1.3.1. The aim of this research

The main aim of this research is to make a contribution to the methodology of understanding urban morphology transition process.

1.3.2. Main objectives:

To develop a CA model that can represent the different trends of temporal-spatial pattern for a case study city in China for different time periods.

1.3.3. Sub objectives:

1. To design a CA model structure to simulate the urban form change process for the study city
2. To define model elements
3. To prepare the necessary data to be used in CA simulation model
4. To implement the CA model prototype and calibrate the prototype
5. To represent the different developing trends in two different time periods
6. To validate the established CA model

1.4. Research questions

From the above stated objectives the following research questions arise:

1. *To design a CA model structure to simulate the urban form change process for the study city*

- How to define the model structure in which the main factors determining urban form change can be incorporated?
- What are the main factors determining the change of urban form?

2. *To define model elements*
   - How to define model elements in terms of cell states, transition rules, and lattice and time steps?

3. *To prepare the proper data to be used in CA simulation model*
   - Which kind of data does this model need? In what scale?

4. *To implement the CA model prototype and calibrate the prototype*
   - How to adjust the model using the real data from field?
   - What is the relationship between the model time and real world time? How to correlate them?
   - What kind of modification on basic CA model needs to be done to make the model able to represent urban dynamics better?

5. *To represent the different developing trends in two different time periods*
   - How to incorporate the different trends of urban development in different periods into the model?

6. *To validate the established CA model*
   - How correct is the simulation versus actual development?
   - How to validate the CA model?
   - Which aspects of the CA model can be changed to improve the resemblance based on the validation?

1.5. Methodology

First of all, the driving forces of urban growth have to be identified. This task will be done by the combination of literature review that can give a general overview about it and field work for investigation. To work on empirical data, the fieldwork plays a very important role in this research. The factors derived from literature review can only give some general or common accepted influences on urban growth. While in the study city, situation may not or possibly may not be the same as the cities studied in literatures. The fieldwork has to reveal the particular growth pattern and key factors that influence the urban growth in the study city.

In the second step, the data requirements will be defined and data will be collected accordingly. At the same time, from literatures, the suitable CA model structure that is closer to the study purpose and is likely to be adopted in this research should be chosen. The model components should also be identified during this stage. The data processing will start in step 2 and end in step 3. (See Figure 1)

In the third step, the conceptual model structure and model components will be defined.

In the last two steps, the real data from case study city will be used to implement and calibrate the CA model. Scenarios will be generated from the calibrated CA model.

Further more, the validation of the CA model is also very important because it gives an impression of how reliable the scenarios are and how good the model match to the reality. While the main purpose of this research is on the construction of CA model, and there is a time limitation, the validation of this CA model will be done using a simple approach.

Error! Not a valid link.

Figure 1 Research methodology

1.6. The outline of the theisis

Chapter 1 briefly introduced the background questions and purpose of this research as well as the methodology used in this research.

Chapter 2 is an overview of both the concept and development of Cellular Automata in urban modelling context. Particular attention is drawn to how the CA model should be modified to reflex the urban dynamics. Multi Criteria Evaluation technique is also introduced in this chapter for their usability in CA urban modelling.

Chapter 3 introduces the study area, Zhongshan city in south China. The basic natural conditions and social economic development in Zhongshan are briefly described.

Chapter 4 is the methodology part. In this chapter, the influencing factors in urban development are identified and the ways to represent these factors in CA model are defined. Finally, the conceptual CA model for study area is developed together with its components definitions.

Chapter 5 is the implementation part. From the software/platform selection, implementation, calibration to evaluation, this chapter covers all the operational aspects of CA urban modelling process. Data from field are used to generate scenarios.

Chapter 6 discusses on issues encountered in this research. Recommendations as to the future development of CA urban model are given. Finally, the conclusion on this research is given in this chapter.

In appendices, the detailed but auxiliary information will be given on raw data, the alternative CA model design, the developing tips, source codes, etc.

2. Cellular Automata and Urban Modelling

This chapter provides an introduction of the theory of Cellular automata and its applications in urban simulation field. Starting from the basic principle of Cellular Automata, the characteristics of CA and why it is widely used in urban simulation field and how its definitions were relaxed to fit in the urban simulation realm were introduced. Then, the taxonomy of urban CA models was given and the current CA urban model developments were introduced. Finally, the concept of multi-criteria evaluation and its applications in urban CA models are given.

2.1. Cellular Automata models in urban study

Cellular automata (CA) have been applied to many fields and gain a more and more important role in urban study realm recently. Due to its flexibility and usability in modelling complex world, urban researchers employ it to simulate the developing processes of cities, exam hypothesis and explore the urban dynamics. The concept of CA, its characteristics, suitability of applying to urban simulation, its advantages and its disadvantages in this field will be given and applications using CA to simulate urban dynamics were introduced. Several questions are to be considered:

- What are cellular automata? How does it work?
- What are the specific characteristics of CA? Why do we use it in urban simulation?
- What do we need to modify on the classic CA when we use it in urban field?
- What are the results of previous CA researches in urban simulation?

2.2. Basic cellular automata

2.2.1. Origin of cellular automata

CA was first devised by John von Neumann and Stanislaw Ulam in 1940s as a framework for investigating the logical underpinnings of live (Rennard, 2000). Due to a lack of high performance computation capacity before the 1970's, Cellular automata was only limited studied and used by few researchers in a pilot research environment for a long time until 1970 when John Horton Conway (Gardner, 1970) made his famous "Game of Life". John H. Conway came up with the "game of life". This simple Cellular Automata was able to demonstrate quite complicated dynamics, which look much like those of live microorganisms. Though the original aim of Conway's life is entirely mathematic, the game's lifelike nature, with its endless complexities and unpredictability's, has captured the imaginations all over the world. The success of Conway's Game Life stimulated the subsequent studies of CA. From then on, extensively researches in CA have been done due to the rapid growth of computation capacity.

2.2.2. Principle of cellular automata

Definition:
"An automaton essentially comprises a finite state machine that exists in some form of tessellated cell-space" (Torrens, 2000). Cellular automata are - by definition - dynamical systems, which are discrete in space and time, operate on a uniform, regular lattice - and are characterised by "local" interactions. "Cellular automata are computable objects existing in time and space whose states change discretely and uniformly as a function of the states of neighbouring objects (cell). What is essence is that CA cells change their states in space and time only locally, namely according to the cells which are strictly adjacent to it."(Batty, 2000)

According to (Torrens, 2000) and (Itami, 1994; Wolfram, 1984), CA has 5 primary elements:

- Lattice: the space on which the automaton exists
- Cell: in which the automaton resides and thus constrained its states
- Neighborhood: the cells surrounding the automaton
- Transition rules: the behavior of the automaton
- Temporal space: discrete time steps in which the automaton evolutes

These will be further referred to as CA-elements. A typical cellular automata start with a one- or two-dimensional lattice of cells or sites. In lattice, cells are identical to one another. In theory, the lattice is wrapped around itself and represents infinite space. This wrap forms a circular in one-dimensional CA and a torus in two-dimensional CA. In practices, it is useful and meaningful to limit the space to a finite space (Itami, 1994). Every cell in the CA has a single value or state out of a finite value range. The simplest range of the states value could be as simple as 0 and 1. Each cell changes its states in the next time step according to the states of its immediate adjacent neighbours (neighbourhood). The function of the neighbourhood states, which is used to change the current cell state, is expressed as transition rules. In the sequence of time (t, t+1, t+2…), each cell in the CA lattice update their states based on the transition rules at the same time.

The general definition in mathematical notation is:

$$\{S_{t+1}\} = f(\{S_t\} \bullet \{I_t^h\})$$

Where $\{S_{t+1}\}$ is the set of all states of the cells in the CA, and $\{I_t^h\}$ refers to the set of all input neighbourhoods. $f()$ describes the functional relationship, t is the time steps in temporal space, h is the neighbourhood size.

The states for each cell in CA is determined by the following notation:

$$s_{it+1} = f(s_t, I_j^h)$$

Where s_{it+1} is the state of a given cell at time t+1, and s_t is the state of that cell at time t. I_j^h is the input of neighbourhood (size h) in the vicinity of cell j at time t influence cell state transition in the next time step. (Torrens, 2000)

Or it can be defined as this:

***Q* = < *S, N, T*>**

Where ***Q*** is the global state of the CA system, ***S*** is a set of all possible states of the automaton, ***N*** is the neighbourhood of all cells that provide input values for transition function ***T***, and *T* is a transition function that defines the change in state of the cellular automaton from its current state to the state of next time step (t+1) (Itami, 1994).

One-dimensional CA:

This is the simplest CA form. Given a table with only one row:

0	1	1	0	0	1	0

The lattice is wrapped which means the right neighbour of the most right cell is the most left cell and *vice versa*. The neighbourhood definition is the immediate adjacent cells (the box). At the beginning, the initial state of each cell was assigned randomly. The state of each cell of subsequent generation is determined by the state of itself and the states of its neighbours by applying the transition rules. We can set the transition rules like this:

1. If a cell is dead (value equals 0) and its neighbors are all dead, it remains dead.
2. If both a cell's neighbors are alive (value equals 1), the cell dies or remains dead.
3. If a cell is dead and exactly one of its neighbors is alive, it becomes alive.
4. If a cell is alive and its left neighbor is alive, it dies.
5. If a cell is alive and its right neighbor is alive then it survives.
6. If a cell is alive and its neighbors are dead then it remains alive.

Using these transition rules, the next generation is:[1]

1	1	0	1	1	1	1

Wolfram has discovered that only four classes of CA behaviours emerge from the one-dimensional CA (Wolfram, 1984).

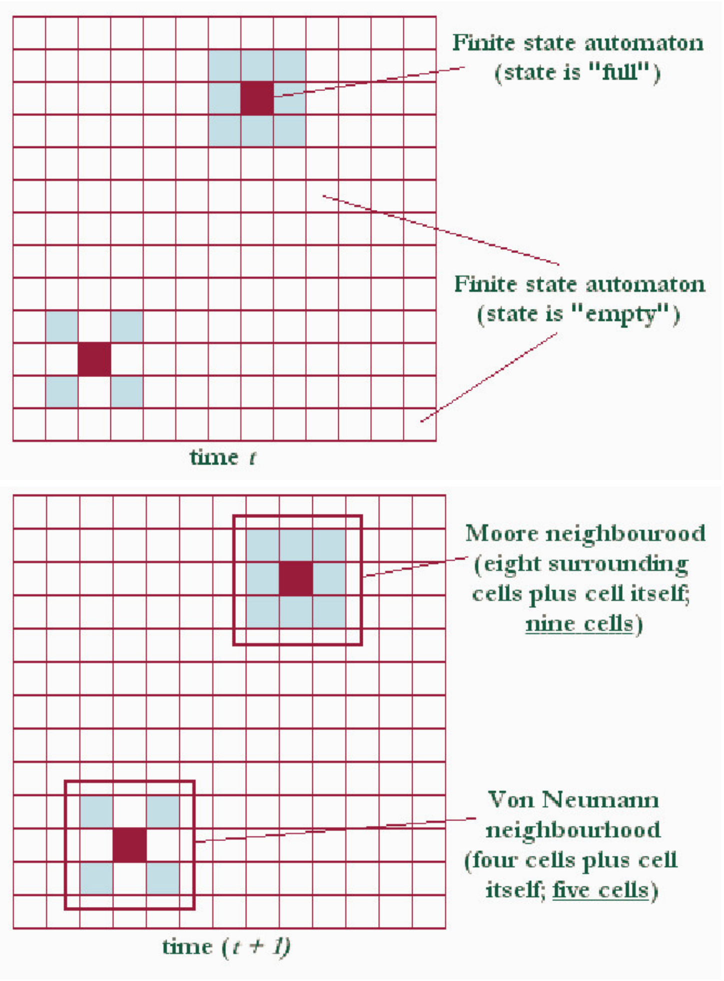
1. Class 1: Evolution leads to a homogeneous state after a few generations in which all cells take on the same state value.

[1] Note that the new generation is generated using the cell states in generation one; the new states of these cells will not affect its neighbours in this generation as a sequential effect.

2. Class 2: Evolution leads to limit cycles (a set of separated simple stable or periodic structure cycling forever in a sequence of states).
3. Class 3: Evolution leads to chaotic aperiodic structure or pattern (not random but without regularity in space and time).
4. Class 4: Evolution leads to complex structures, some of them are very long lived, some propagating.

Two-dimensional CA:

Two-dimensional CA just simply extends the lattice of one-dimensional CA in an additional dimension. The automata are distributed in a two-dimensional lattice.



The neighbourhood include the cell itself (dark brown one) and the light grey cells.

Figure 2 Two dimensional Cellular Automata (Torrens, 2000)

There are two basic types of neighbourhood definition, Moore and Von Neumann neighbourhood (as shown in Figure 2). For different purposes, researchers have tinkered the template of neighbourhood with size and configuration into various types of neighbourhood.

For two-dimensional CA, the most famous example is Conway's Game of life. This game is a mathematical game, which is played on a two-dimensional lattice. The neighbourhood of Conway's Life was specified as Moore neighbourhood. Each cell of the lattice has only two states, live-1 or dead-0. The state of each cell in the next time period is dependent on the status of itself and its eight nearest neighbours in the current time. Conway used three simple transition rules to control the states of each cell in his game space. Following is the transition rules:

1. Survivals: every live cell with two or three live neighbors survives into the next generation.
2. Deaths: every live cell with four or more live neighbors dies for crowd and those less then two live neighbors die for lonely.
3. Births: every dead cell gets alive if they have exactly three live neighbors.

At the beginning of the game, some live cells with 1 state were randomly seeded in the lattice. When the game runs, these transition rules are applied to each cell in the cellular space. The states of the cells were displayed on the screen. These simple rules led to enormously complex patterns, which show some lifelike traits. The publication in the public domain of Conway's game of life triggered many subsequent studies on CA and itself becomes a pivotal point of CA development.

Complex characteristics of CA:
From very simple rules, CA can generate tremendously complex patterns, which is naturally tied to complex systems.

Self-organization:
From Wolfram's work, CA shows a self-organization characteristic. Even from a random or very disordered initial structure, after many iterations, which, also called evolutions, it can reach a quite organized state which shows some special pattern (Wolfram, 1983).

Self-similarity and fractal dimension:
Many natural systems, including snowflakes, appear to exhibit fractal patterns. It is possible that in many cases these fractal patterns are generated through evolution of cellular automata or analogous processes. The experiments done by Wolfram have shown that it is quite often that CA models using simple transition rules finally end up with self-similarity and fractal dimension characteristics (Wolfram, 1983).

2.3. CA in urban modeling

"Urban models are an abstraction, simplified versions of real world objects and phenomena that are used as laboratories for exploring ideas about how things work in cities. CA is no exception to this characterization."(Torrens, 2000)

CA theory was first applied as a growth modelling technique in the biological sciences. Influenced by the scientific method in their thinking, urban planners brought CA theory to urban modelling with promising possibilities (Coucleis, 1997). CA is believed to be applicable to any natural system. Cellular-spaced modelling is characterized by homogeneity and uniformity of structure because the state of a cell is determined by both the state of its neighbouring cells and defined growth/death/birth rules that are applied uniformly without exception. The rules, which control the model behaviour, are uniformly applied to all the cells populated in the cellular space. It can also be explained as all the cells in the cellular space use the same set of rule to determine their next states. Whilst the homogeneity and uniformity of structure can be modified in CA models by adding input variables and defining the set of neighbours to cells that are not geometrically adjacent. If this is done, then the behaviour of cells is no longer uniform, and each cell or group of cells requires their own unique transition functions(Zeigler, 1984). For instance, a particular land use development in a city could have different developing preferable requirement in terms of the accessibility or land prince. This can be represented in CA model as different transition rules for different land use classes. Another example could be the different consideration of neighbourhood by different land use type. In this sense, the behaviour of the CA model is no longer uniform.

2.3.1. Complex world and urban dynamics

"A complex system is a system for which it is difficult, if not impossible, to restrict its description to a limited number of parameters or characterizing variables without losing its essential global functional properties"(Pavard and Dugdale, 2002). Formally, a system starts to have complex behaviours (non-predictability and emergence etc.) at the moment it consists of parts interacting in a non-linear fashion. It is thus appropriate to differentiate between a complicated system (such as an air-plane or computer) and a complex system (such as ecological or economic systems). The former are composed of many functionally distinct parts but are in fact predictable, whereas the latter interact non-linearly with their environment and their components have properties of self-organization, which make them non-predictable beyond a certain temporal window (Pavard and Dugdale, 2002).

Complex systems have a number of properties, some of which are listed below:

- Non-determinism and tractability
- Limited functional decomposability
- Distributed nature of information and representation
- Emergence and self-organization

The first one means it is impossible to precisely anticipate the behaviour of such systems even if we completely know the functions of its constituents. The second one implies that it is very difficult to decompose the complex system into functionally stable components. The third property means the system, as a whole is not equal to the sum of the functions of all its parts. The last property is the key property of a complex system. Based on the interaction of components of the complex system, some behaviours and patterns emerge as a result of the patterns of relationship among the elements. It is a global phenomena that cannot be anticipated even the components of the complex system have already been understood thoroughly.

As Pavard and Dugdale argued, a truly complex system would be completely irreducible which means it is impossible to derive a model to represent the real system without loosing some of the properties of the system. Yet researches in physics and to some extent biology and other fields have shown that the basic components of many systems are quite simple (Wolfram, 1988). Many observations show that from these very simple interactions within some simple components can produce extremely complexity in those physical or biological systems in terms of form and dynamics. The main idea of complexity study is emergence. In an emergent complex system, a small number of rules or laws, applied at local level and among many entities are capable of generating surprising complexity and often ordered patterns in aggregate form. Yet since the complexity is created by the relationships amongst elements in the system, no single element can be distinguished to be the main or dominant factor in the system. A corollary of this is that no element in the system could hope to control the system. Often the complex system can be very much diverted by some intermediate perturbations.

Urban dynamics as complex system

Despite many natural systems tend toward disorder when evolving with time, a large class of systems, biological ones being prime examples, show a reverse trend: they spontaneously generate structure with time, even when starting from disordered or structure less initial states (Wolfram, 1983).

The city is a very good example of a complex system, where the parts can only be understand by the whole and the whole phenomena is more than the simple sum of the individual parts. Many components act on each other within the city, land use, transportation, culture, population, policies, economics and so on. The relationships between each other are neither clear nor linear. Spontaneous and simultaneous individual decision-making processes take place both in spatial dimension and temporal dimension (Barros and Sobreira, 2002b).

2.3.2. Relaxation of classic CA to use it in urban models

Whilst CA models shows high potential usability and suitability in modelling urban dynamics, the basic definition of CA by Wolfram and Conway cannot be used immediately in this field without some extent of modification. Typically a CA model is a close system, which means no information, or energy or mass is exchanging with its outside world while it is not the case in an urban context. Any city, or complex system, is an open system which exchanging energy and/or mass with its environment. The classical CA definition is too strict to accommodate all the commponents need to represet in a CA urban model. Thus, modifications on classical CA are inevitable.

The modifications have to be applied to almost every element in CA model to make it capable of representing urban dynamics. Conclelis (1985 cited by Itami, 1994)) proposed a generalized CA (GCA) model to relax the basic CA limitations to situate CA in to urban simulation realm (Itami, 1994). It can be summarized as following:

- The space could be open and finite and the “cells” could be irregular. (Local components)
- The neighborhood could be defined in different ways for each “local component”. It does not need to be the adjacent “cells”.
- The transition rules may be different for each of the cells and these rules can be applied in different time intervals.
- Both input and output from the simulation model may be defined.

To be able to use in urban simulation, classic CA has to be modified, mainly to relax the restrictions on the CA elements.

- Cell space:

In basic CA, cell space is a close system. It does not accept the information from outside. While in urban context, one cannot look the city as a close system because so many exogenous links exist. To incorporate these exogenous influence into CA, urban modellers often use constrains and algorithms applied to transition rules.

Some recent researches tried to use several concurrent states in a variety of forms instead of the homogenous states which are in the same form.(Torrens and O'Sullivan, 2000) This modification introduced a great flexibility into CA model. Some researches have divided the cell states into two groups, "fixed" and "functional". These applications treat those attributes that are not changed in the urban development process as fixed states, such as watershed and terrain. Meanwhile, sites that are changed or active in the developing processes are considered as functional which can evolve with time, such as population and land use.

- Lattices

The lattices in classic CA are defined as an infinite uniform space of regular grid. While in the urban context, the space of development is limited either by natural constraints or by policies. The typical CA space configuration is a continuous space or "torus" which means its edges are connected to the opposite edges and the cells in CA model are identical in their geometry. In urban context, it is not realistic to use a wrapped space to present the real city and it is possible to use spatial meaningful unit such as parcel or road segments as "cell" to present the urban unit. The spatial unit may be different either in size or in shape. When simulating urban dynamics, the lattices of CA are often limited in a finite extent which presents the physical area of the city and the shape and size of CA cells are also be modified some times like the model proposed by O'Sullivan which use a graph cell form (O'Sullivan, 2000).

- Neighbourhoods

The neighbourhood of a cell in the CA formalism consists of an individual cell itself as well as a set of adjacent cells. There are two types of neighbourhood in strict two-dimensional CA, Von Neumann neighbourhood and Moore neighbourhood. (See Figure 1) This definition is not able to incorporate the action-at-a-distance although the action can propagate indirectly as the model evolutes. In urban development, not only the adjacent properties influence the current cell (site) states, but also the city-wide actions. To adopt this type of action (citywide), modellers modified the neighbourhood template configuration a lot to represent the action-at-a-distance. They use weights applied to neighbourhoods in transition calculation to represent the distance decay effects and use extend neighbourhood to comprise larger spaces.

- Time

In classic CA, time is discrete and the cells are updated synchronously between time steps as transition rules are applied simultaneously at every location. Recently, researchers tried to asynchronous update cell-state via the actions of agents in a CA space. Thus come some hybrid CA-Agent models. Barros and Sobreira (Barros and Sobreira, 2002a) proposed a hybrid model in this flavour. This type of model is under extensively studying recently. Torrens also proposed a hybrid model which combines the CA and agent-based model (Torrens, 2001).

- Transition rules

The essence driving force behind the CA model is the transition rules. It translates the behaviour of the real world into CA models. In classic CA, transition rules are deterministic and will not change during the evolution of CA. While in urban simulation, transition rules have to be opened to exogenous effects and they are modified into a probabilistic expression. This introduced an element of randomness, or called “noise”, into the model. Also it can be made dependent on other rules within the model to reflect the idea that urban systems operated as a tangled web of co-dependent subsystems and phenomena (Torrens, 2000). In Yeh’s model, the transition rules introduced a stochastic disturbance into the model to represent the haphazard development in urban fabric (Yeh and Li, 2001). In this case, the “noise” was introduced into the CA model. Other researchers have used a probability function to control the model action, like the work of Batty ((Batty et al., 1999) , the action of transition rules is a contingent action upon a certain probability.

In urban simulation CA model, the transition rules also have been relaxed or modified by using a fuzzy logic controlled method (Wu, 1998; Yan and Phinn, 2002). The introduction of fuzzy logic control into CA model enable CA model to capture the feature of land conversion behaviour which is naturally described as fuzzy terms such as fast development or slow development.

2.3.3. Main types of CA in urban simulation

According to the purpose of a CA model, they are generally divided into three not mutually exclusive categories:

- Models to explore spatial complexity
- Models to test theories and ideas in urban context in a abstract manner
- Models as operational urban planning support systems

The first one is to advance understanding of complex systems (cities as we discussed before are no-doubt one candidate) while the second one is to examine the roles or factors in the complex system as a driving force in urban dynamics. An abstract urban-theoretical model was used as a “social laboratory” for testing ideas or theories about city. The last type of CA urban models is mainly used as part of decision-making support system and serves as scenarios-generators. As Torrens argued, only very few of the third type of models which categorize as operational model were really operational (Batty and Torrens, 2001; Torrens, 2000).

These three different kinds of applications differ very much in terms of modification on classic CA model. For those CA models used to explore the complexity of cities, with very less modification of the classic CA, they are able to achieve the exploring goals. While those models proposed to be used for practices in urban planning and decision making processes have to modify the classic CA model a lot to generate some extent of resemblance of reality.

Based on the above discussion, we can take a look at real cases implemented earlier and the idea of those CA models currently being proposed or implemented.

For the first category of Torrens' taxonomy, exploration of urban complexity, Wu (1998) had integrated GIS and CA together with a multi-criteria-evaluation technique to model the growth of cities. The role of GIS is to provide the initial state and to enable the measurement of factors constraining the land development (Wu and Webster, 1998).

For the second one, theory and hypothesis testing, there are many cases belonging to this category. Batty (1998) has built up a simulation environment to simulate urban evaluation using extended cellular automata. In his model, the main purpose is to examine theory and hypothesis (Batty, 1998). Batty and Xie et al. also built up a GIS-based cellular automata model using probabilistic transition rules to study the urban developing theory (Batty et al., 1999). Besussi used cellular automata examining two hypothesis of "diffused city". In his research, a family of cellular automata models is developed in order to investigate the diffusion processes (Besussi et al., 1998).

For the last type of CA, operational models, Martin and Wu had implemented one application of a CA simulation to urban growth using empirical data for the southeast region of the UK. They discussed the trade-off between empirical and theoretical considerations when implementing CA simulations, and the nature of the empirical constraints that may be applied (Martin and Wu, 1999).

2.4. Current developments of CA urban models

CA models are already widely applied in the urban simulation realm. The pioneer researchers intensively explored both the theory and practical aspects of CA models.

As O'Sullivan and Torrens argued(Torrens and O'Sullivan, 2000), there are several aspects of the possibilities of putting effort in:

- Strict formal CA with a small family of geographical process rules.

This mainly for those models used to explore the urban dynamics theoretically. With very less modification of formal CA, the behaviour of the urban model can be explored thoroughly. Segregation, diffusion, growth, etc, process have been already modelled into this kind of CA models.

- Cellular models with irregular lattice structures

This idea comes from Takeyama and Torrens (O'Sullivan and Torrens, 2001; Takeyama and Coucleis, 1997), while it was clearly expressed by O'Sullivan in his article that a graph-based CA model was proposed to be used (O'Sullivan, 2000).

- Agents in cellular models

Human agents like developers, firms, financers etc, manoeuvre, collaborate and change the form of city according to their own purposes. To translate these human activities into CA transition rules is not a straightforward procedure. It seems plausible to use agents to represent this type of activities. O'sullivan suggested that agent-based approaches maybe more appropriated sometimes.(O'Sullivan and Torrens, 2001). This point is typically a suggestion that we should use a hybrid model structure that combines the CA and agent-based approaches. Cecchini and Rinaldi (Cechini and Rinaldi, 1999) also proposed a multi cellular automata structure that use two or more CA models to incorporate more complicated phenomena. In their model configuration, two or more CA models linked together in sequence, the output of one CA model is the input of another CA model.

- Asynchronous cell update

Some researchers also suggested that the synchronous update of all cells in the same time step is not sound realistic in urban dynamics. Different sites and different locations have their own developing rates that each iteration of it does not mean the same time interval. How to synchronize this in the same CA model is still problematic.

2.5. Multi-criteria evaluation in CA urban model

Multi criteria evaluation could be used in any case involves decision-making or classification. As defined, the notion of evaluation is conceived as attempting to achieve the objectivity of a certain (choice) situation and the appraisal of that situation.(Voogd, 1983) Multi-criteria analysis establishes preferences between options by reference to an explicit set of objectives that the decision making body has identified, and for which it has established measurable criteria to assess the extent to which the objectives have been achieved. In simple circumstances, the process of identifying objectives and criteria may alone provide enough information for decision-makers. (Department for Transport, 2003)

2.5.1. The basic principle of MCE

First a matrix of evaluation should be constructed and its elements reflex the characteristics of a given set of choice possibilities. The choice possibilities are determined by the criteria (Voogd, 1983). A criterion is some basis for a decision that can be measured and evaluated. It is the evidence upon which the decision is made.

Table 1 An score Matrix

		Choice possibilities			
		Choice 1	Choice 2	Choice 3…..	etc
Criteria	1 2 3 . . Etc.	Criteria scores			

Voogd called this matrix “evaluation matrix”. If the score has equal unit or standardized, it is called “effectiveness matrix”. This matrix simply shows the measure of each criterion under each choice and the measures are usually in different units(Voogd, 1983).

To evaluate the choices in evaluation matrix, the various view concerning possible criterion priorities is summarized as a Priority matrix:

Table 2 Priority matrix

		Criteria			
		1	2	3…	Etc
Views	1 2 3 . . Etc.	Priorities			

In table 2, the viewpoints of different stakeholders involved in the decision making process are listed at left hand. From the viewpoint of each stakeholder, one set of priorities for each criterion is given. The priorities can be given by means of quantitative number, which is often referred to as a weight system.

Combining score matrix and priority matrix, by applying arithmetic technique and priorities, another matrix, appraisal matrix could be derived. This step is where the weight system taking place.

Table 3 Appraisal matrix

		Choice possibilities			
		Choice 1	Choice 2	Choice 3…	Etc
Views	1 2 3 . . Etc.	Appraisals			

Table 3 Appraisal matrix gives an indication of the general quality of the choice-possibilities under different consideration, viewpoints. The results of the appraisals evidently depend on the criteria characteristics, the priorities that consisted a weight system, and the arithmetic technique used. (Voogd, 1983)

2.5.2. Scoring and weighting

MCA techniques commonly apply numerical analysis in two stages:

- Scoring

The expected consequences of each choice possibilities for each criterion are assigned a numerical score on preference scale, which is a scale representing relative strength of preference. Usually the most preferred option on a given criterion is assigned the highest score value and the least preferred option on that criterion is assigned the lowest value. More preferred options score higher on the scale, and less preferred options score lower. In practice, scales extending from 0 to 100 and scale extend from 0 to 1 are often used, where 0 represents a real or hypothetical least preferred option, and 100 or 1 is associated with a real or hypothetical most preferred option. All options considered in the MCA would then fall between 0 and 100 or between 0 and 1.(Department for Transport, 2003)

- Weighting

Numerical weights are assigned to define, for each criterion, the relative valuations of a shift between the top and bottom of the chosen scale. It is the representation of the priority of different criteria. All the weights for all the criteria together are called weight system.

2.5.3. Linear additive MCE models

If it can either be proved, or reasonably assumed, that the criteria are preferentially independent of each other, then the simple linear additive evaluation model is applicable. The linear model shows how an option’s values on the many criteria can be combined into one overall value. This is done by multiplying the value score on each criterion by the weight of that criterion, and then adding all those weighted scores together. However, this simple arithmetic is only appropriate if the criteria are mutually preference independent. Most MCE approaches use this additive model.(Department for Transport, 2003)

2.5.4. Compensatory and non-compensatory MCE techniques

Two kinds of MCE techniques are frequently used: compensatory techniques and non-compensatory techniques.

Methods for combining assessments on separate criteria into an overall assessment such that lesser scores for an option on some criteria can be offset by greater scores on other criteria, i.e., trade-offs are modelled are called compensatory MCE approach. Low scores on one criterion may be compensated by high scores on another. For example, assuming a site has low suitability in terms of geological condition for constructing buildings and its has very good accessibility and enough investments, the overall assessment of this site may get a relatively high preference for constructing new buildings since the high scores in accessibility and investments compensate the low score of geological conditions. The most common way to combine scores on criteria, and relevant weights between criteria, is to calculate a simple weighted average of scores.

Another MCE technique is non-compesatory approach. This method does not permit trade-offs between criteria, i.e., poor values for an option on one criterion cannot be offset by better values on another criterion. So it is often used to represent the obligatory preferences. For instance, if an area is conserved by government as water resource area, it is not allowed to be developed into any other kind of land use even the natural condition in that area is very suitable for other land use such as agriculture or industrial. The most common way to represent this approach is to multiply the non-compensatory criteria which are represented by binary values, 0- not allowed and 1-no restriction.

2.5.5. The use of MCE in CA model

In CA urban models, the transition rules are often dealing with some kind of decision which involves many factors especially when modelling the real development of cities. To decide the next state of one cell in a CA urban model, the current cell has to consider its own current state, its surrounding environment conditions, etc. Hence, the MCE methods find their way in CA urban models.

Due to the ease of interpret of MCE, it is widely used in CA urban model to formulate transition rules. Wu (Wu and Webster, 1998)has used a MCE transition rules to control the evaluation of current states of a particular cell, Yeh(Yeh and Li, 2001) has applied a constrained CA model to present the constrained urban development.

2.6. Conclusion

From the above literature, we draw a brief overview of CA model, the principle of CA, the use of CA in urban simulation, the relaxation of classic CA, the current trends of urban CA models. According to the literatures about CA model in urban simulation, most of the applications are about the exploring of urban complexities and infusing the theory of urban growth and development into CA models. The operational models are rare and only very few of them produced some resemblances of the reality. Still many efforts are putting on the operational models by the researchers and they are trying to make CA urban model into a practical urban planning tool.

Hybrid models are under intensively studying recently, which suggest that to achieve to some extent realistic meaning, one should combine the traditional model, CA model and agent-based model. Whilst still some researchers doubt about the heavily modification on classic CA, they believe that this kind of modification loose the theory foundation of CA and make it not acceptable even it can get some plausible results. Still to build a workable, usable operational model for modelling real urban dynamic such as growth and sprawl, one has to move away from the formal simplicity of the classic CA. (Torrens, 2001)

In the following chapters, a CA model to simulate the urban growth process of a middle size city in south China, Zhongshan city, will be set up. This research will focus on the construction aspects of CA urban model and try to build a CA model that is able to work as a scenario generator, which could be manipulated by the researchers and planners. In this sense, it should belong to the second category in Torrens' taxonomy.

3. Zhongshan: the Study City

In this chapter, the situation and development of the study area, Zhongshan city was introduced. First the general descriptions about the location, natural environment and social economic conditions were given then come the urban development from 1980s to 2000s. The social economic change as well as policy changes during these periods were reviewed and were regarded as the driven forces in the formation of urban morphology.

3.1. General Description

Zhongshan is one of the famous cities in South China, and the name of Zhongshan was got from the name of the first President of China -- Dr. Sun Yatsen (Zhongshan) who was the pioneer of Democratic Revolution of China. It was a small city in 1980s (it has a 5.6 square kilometres build-up urban area in 1980) and has experienced a rapid urban expansion and economic growth during last two decades.

It is also famous for the beautiful environment and good living conditions. In 1992, Zhongshan Municipality was honoured as one of the 10 best cities of China in environmental sanitation. In 1994, Zhongshan Municipality was honoured as Advanced City of China in Garden and Plantation by Department of Construction. In 1995, Zhongshan Municipality was honoured as National Hygiene City of China. In 1996, Zhongshan Municipality was honoured as National Garden City of China. In 1997, Zhongshan was awarded as Habitat Scroll of Honour Award by United Nations. This award was the only one in Asia. In 1998, it got another award -- the State Environmental Model City of China.

3.1.1. Location

Zhongshan Municipality locates in the middle southern part of Peal River Delta, in lower reaches of Xijiang and Beijiang Rivers. Panyu and Shunde are adjacent to the north of Zhongshan, Jiangmen, Xinhui and Doumen are adjacent to the west, and Zhuhai is to the southeast. Zhongshan faces Hong Kong and Shenzhen across Lingdingyang of Pearl River Mouth. It is located between 22o11' to 22o46' north latitude, and between 113o09' to 113o46' east longitude (Figure 3). Its total administrative area is 1,800.14 km2.

Figure 3 Zhongshan city (location and administrative boundary)

3.1.2. Natural Environment

The topography consists of mountains arisen from continental shelf, hilly and table lands and alluvial plain of Pearl River estuary. It has a south subtropical monsoon climate with sufficient heat and light and abundant rainfall, and annual average temperature is 21.8^{o}C and annual average precipitation 1,748.3mm. The city mainly located in the flat part of this area.

River system in Zhongshan is composed with two river networks: the river network in plain and the river network in hilly region. Three of the eight Pearl River mouths, Modaomen Mouth, Hengmen Mouth and Hongqili Mouth, flow to the South China Sea by passing through Zhongshan. The main water channels are Jiya Channel, Xiaolan Channel, Hengmen Channel, Huangsha Channel, Huangpu Channel, Qijiang River, Beitai Stream and Dahuan River. The areas of Madafeng Water Plant in Xiaolan Channel, Quanlu Water Plant in Xijiang River and Changjiang Reservoir are classified as first class reserves for drinking water source. The river system is an important constituent of transportation system in Zhongshan city. The urban developments before 1980s are mainly taken place along the Qijiang River and two main roads. River system also gives Zhongshan city extra limitation on the city form. From 1992, the municipality posed strict control on any development that intends to fill some small water body such as lakes, ponds, etc due to the policy changed to emphasize on natural water reservation. The same policy had also been applied to public green area and natural forest parks.

Vegetation in Zhongshan is mainly artificial vegetation and natural evergreen monsoon forest. Forest coverage is 12.95%. There are some forest parks set up in Zhongshan. Reserves for agriculturally ecological environment have been built in northern, western and southern parts of Zhongshan. Within the urban region of the city, a 100-hectare ecological park has been constructed, and coverage of forest and grassland is about 35.96%, and each capita with a 9.39m^2 public green area. In these parks, Zhimaling Park with an area of 87.53 hectare is one of the largest city parks with both urban and ecological functions in Guangdong province. The local government strictly protects these green areas for a sustainable development purpose since 1992.

3.1.3. Economy

Zhongshan is one of the Chinese coastal cities first opened to the world. Population registered as citizen of Zhongshan was 1.3008 million in the end of 1998. All enterprises in Zhongshan Municipality are more than 7500.

The main enterprises are in new high technological industries. There are six groups of industries: new energies, new materials, electronic information, mechanical and electronic products, light textile industry with new high technology, and bioengineering. These industries produced many important products, like family electronic wares (household appliance), fine chemical products, textile chemical fibre, glass building materials, metallurgy machine, electronics, packaging, printing, medicine and foods and drinking. Zhongshan has been one of the most important bases of light industrial products in Guangdong Province.

Agricultural production has been increasing steadily year by year though it is no longer the main economic factor in Zhongshan. City-suburb agriculture and export-oriented agriculture are combined. Seven agricultural production bases have been built and they are the bases of high quality rice, aquatic production industries, livestock farming, fruits and vegetables, fast growing forests, flowers and processing of agricultural products.

The second and tertiary industry in Zhongshan is being developed quickly. Communications, post and telecommunications, water supply, and power supply industries have been greatly developed.

Tourist industry in Zhongshan has been developing quickly, from small scale to well developed large scale. Tourist industry has been increased with a rate of 16% annually over the past 10 years. As a new tourist city, Zhongshan has been called as "a beautiful place with famous people, and green hills and forest, a small city characterized with gardens of South China, and a rich hometown of overseas Chinese".

3.2. Urban Development and Construction

Since reforming and opening to the world started, Zhongshan has become a modern city of middle size. Before that time, it was a real rural region with some little towns. Now agriculture has not been the leading part of industries. Many other industries have been developed in past 20 years greatly. Much of rural regions have been urbanized. Now Zhongshan is already a modern city. Figure 4 shows the administrative boundary (town level) and the water body (blue areas) and built-up areas in Zhongshan city in 1998.

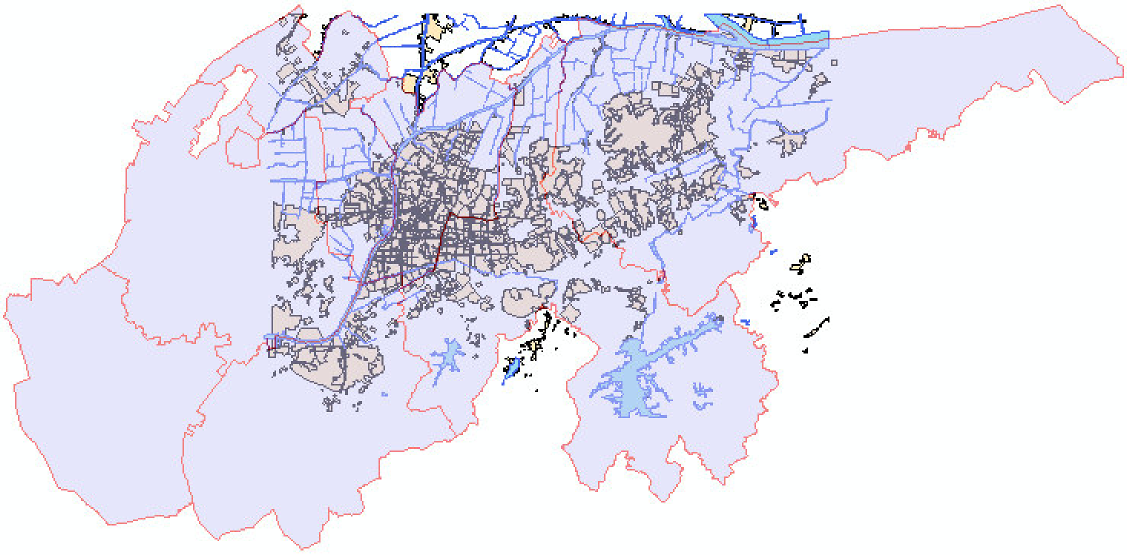
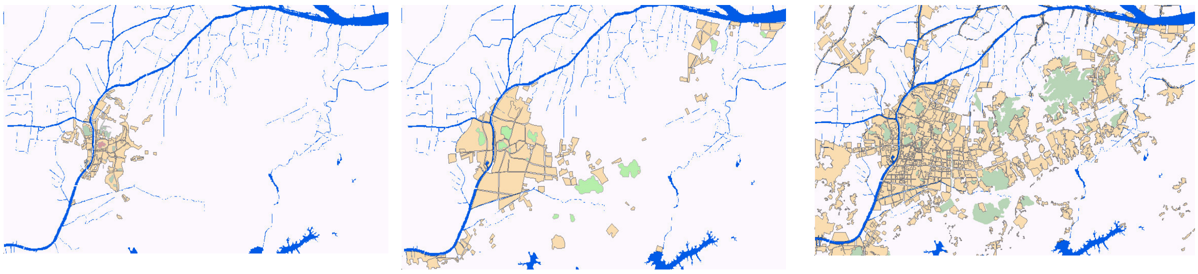


Figure 4 The administrative boundary of Zhongshan

The development can be easily seen on the maps:



Urban form in 1992

Figure 5 Urban form developing during last 20 years (the brown-colour represents urban area)

The current city centre is about 26.59 Km^2 . The area of city centre was 5.6 Km^2 in the early of 1980s(Figure 5). Government of Zhongshan Municipality has insisted on following principles while they made decision for every development: high starting level for planning[2], high standards for constructing, and high efficiency for managing. A master planning for the period from 1992 to 2000 had been outlined in 1993. Under this master planning, many detailed planning of region and industries had been completed, like planning of settlement regions, planning of Torch Development Zone, planning of villages and towns, and planning of Cultural Walking Street in west of Sun Yat-sen(Zhongshan) Road, and so on.

Up to 1997, more than 15 million M^2 building for residents had been completed. A Settlement Engineering had been finished, and it included some large-scale settlement regions. Living space per resident reached about 20.2 M^2. 974 Km highways were built. Its density was about 58 Km per 100 Km^2 which is quite high compare with the average road density in other China cities.

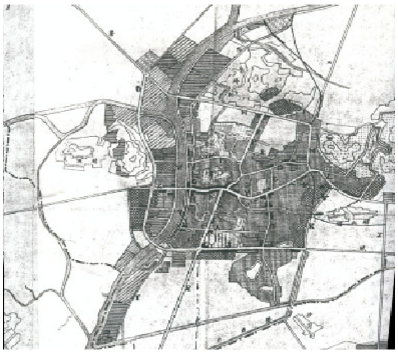
[2] It is in terms of modern planning concepts such as environmental friendly, sustainable development, etc.

3.3. Policy and master plan change

Zhongshan is a historical city developed from a town, it has many old buildings need to
Developed from a relative small city in 1950s, 7.7 square kilometres of built-up area in 1979, Zhongshan kept a quite stable pace of growing until 1990s. The city expanded rapidly in last decade. From then on, the private business played very important roles in the urban development. There also exists a significant change of the way of thinking about urban planning. Before that, the city was design to develop under the strict control of government according to the master plan. While after that, market economy became dominant in southeast coastal cities in China including Zhongshan. Private investors from domestic and abroad came and founded their businesses.

From 1980s, every five or six years, the planning bureau of Zhongshan carried out a new master plan according to the current and desired situation and policies. Some adjustments on the plan during implementation are also regularly carried out. From the master plans, one can see the dramatic change of way of thinking in urban development during the last two decades. (See Figure 6)

Starting from a classical compact mono-centric city form, Zhongshan government changed its development policy in 1985. In the 1985 plan, the planning bureau proposed a multi-centre urban form for future development. The suggested development was along the Guangzhu main road and Shiqi River with several satellite towns surrounding. In 2002, the city has developed into a two-centre form: main centre and Zhongshan harbour centre. These two centres were connected together formulated the strap city form.



Master plan in 1980

Master plan in 1985

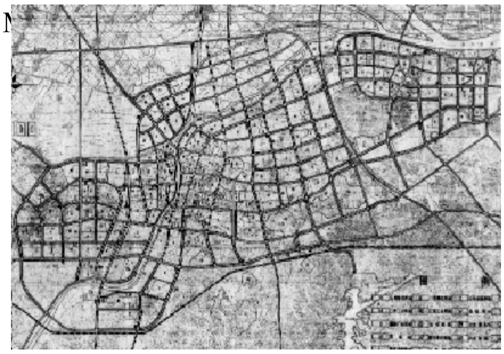


Figure 6Mater plans in 1980, 1985 and 1992

In master plan 1980, the municipality mainly considered the reconstruction of the old urban centre area to improve living condition of residents. Other focus points were more functional structure of industrial distribution and environment protection and agriculture land protection. In master plan 1985, the municipality had already considered the industrial pollution and environment impacts. This master plan suggested that those heavy-pollution related industrials should be moved out the urban centre or banned. The guidance of the 1985 planning still focuses on developing Zhongshan city into a mono-centric city.

In master plan 1992, a dramatically change had taken place. The municipality planed to develop Zhongshan City into a two-centre strap structure based on the main road system and highway frame. The function of the old city centre remains as commercial, residential centre, traditional industries were planed to move to the fringe of the urban area. The new urban centre, Zhongshan Harbour district, was planed to be the new modern high-technique industrial centre. The two centres were to be linked by the development of urban land along the two main roads which link the two centres together. More strict policies on prevention of natural environments and water resources had carried out and the municipality put more effort on building an ecological sustainable city(municipality, 1993).

While the subsequent development did not follow exactly what the master plan 1992 supposed to be. Due to the over estimation of the land demand, many land parcels were transferred into urban land use but with fairly low usage. Many constructions of residential area and industrial area stopped before they could finish due to the lack of following investment and the low expectation of future market. This situation was reflex in master plan adjustment 1998. In this plan, the transition of rural land to urban use in the fringe area was strict limited to control the over expansion of the city. It put more consideration on building an environmental ecological sustainable city.

3.3.1. Conclusion on data regarding use in CA model

Mainly four kinds of spatial data are presently available: Land use maps, master plans, land price, and topographic maps. Due to historical reason, no detailed land use map exists before 1996. Land use maps in 1980 and 1985 and 1992 are in analogue format on paper media and quite general.

Since the water body and terrain changed very less from 1980 to 1998, the water map of 1998 is used to represent the water environment during these two decades.

But the transportation network has changed a lot. The network in 1983 will be used as the starting network and the network in 1992 will be introduced into the model during the running of the model according to the results.

Table 4 are the data used in this research.

Table 4 Data source

	Land use->urban	Terrain	Water body	Master planning	Transportation net work
1980	Land use map (1980)	1983 topographic map	1983 topographic map	1980 master planning	Land use map or 1983 topographic map
1992	Land use map (1992)	1998 topographic map	1998 topographic map	1992 master planning	1992 land use map or derive from 1998 topographic map
1998	Land use map (1998)	1998 topographic map	1998 topographic map	1998 master planning	1998 topographic map

4. Development of MCE controlled CA model for urban growth

In this chapter, a conceptual CA model controlled by MCE transition rule is developed. First, the affecting factors of urban growth are identified based on the literatures and field observations introduced in chapter 3. Then the conceptual CA model design including structure developing, elements definition and transition rules were set up.

4.1. Urban growth and urban sprawl

Urban growth indicates a transformation of the vacant land or natural environment to the construction of urban fabrics including residential, industrial and infrastructure development, it mostly happened in the fringe urban areas. Urban expansion is driven by population growth, social economic development. (Shenghe and Sylvia, 2002) It has always been the focus of the social economic debates due to its impacts on social economic aspects as well as the impacts on the huge consumption of cultivated lands and damage to the environmental sustainability.

China has experienced rapid urban growth since its economic reform. Especially in the end of 1980s and the beginning of 1990s, there was a so-called "great mass fever" on real estates development and establishment of development zones. Accompanying this, the population migration also increased dramatically. In 1980, China's urban population was 191 million corresponding to 19% of the country's total population, and in 1998, it had increased to 397 million. Almost 30 percent of the total population has been classified as urban population at that time. (Yearbook, 1999)

Typically, population densities, soil and climate conditions, and economic development are significantly different across regions in a large developing country like China. China can be perceived as a group of co-evolving, dissimilar economies rather than a homogenous entity.(Klaus and Laixiang, 2001) Hence, the attempt to develop a universal CA simulation model that could be applied to every city is not realistic. The CA model for simulating urban development for a specific city is usually an Ad hoc one.

The study city Zhongshan has experienced a dramatic expansion during the last two decades. The urban development not only took place in fringe-urban areas but also developed towards east sub centre, which is emerged from 1990, along two main roads. This gives some clues that transportation network and urban centres might play important roles in the urban developing process in city Zhonghan. Later in the implementation and calibration part (Chapter 5), this will be used as an initial high weight setting to these two factors.

4.2. Factors affecting urban form

4.2.1. Driven forces

Shenghe and Sylvia's (Shenghe and Sylvia, 2002) study about two major cities Beijing and Shanghai revealed that six kind of urban growth driving forces or factors were identified by literature and primary data analysis. These are:

1. Development of urban economy
2. Increase of investment on fixed asset such as utility network, roads, etc., including foreign investments
3. Increasing income and living standard in urban areas
4. Growth of population and urbanization, especially rural immigrants to urban areas
5. Relocation of industrial enterprises
6. Developments of Town and village enterprises (TVEs)

The first three factors are similar to the driven forces in western counties. They are necessary for any urban expansion process. The growth of urban economy, more investment on utility facilities, increasing income and living conditions would naturally claim more urban land use.

The last three driven forces have to be interpreted in China context. Despite the strict control policies on population immigration, most of the southern China cities including Zhongshan have experienced a rapid growth of rural urban immigration. The relocation of industrial enterprises is mainly motivated by the economic law of famous bid-rent model in urban land market since the economic reform. In this theory, the city center has the highest land price due to its good services level and its good access to facilities. From city center to urban fringe, the land price decrease. Based on this theory, industrial enterprises, which usually concern more on road accessibility rather then concern on facilities and services, could shift from the center area to outside the city and reduce the costs for land investments. Many industrial enterprise located in urban central district in the former administrative system have been moved to the fringe areas gradually. The private enterprises and TVEs now play a very important role in China's economy. These enterprises comprise the major part of Zhongshan's economy.

4.2.2. Factors affecting urban form

As motioned above, the urban growth driven forces, namely the development of urban economy, the increase of investments, the increase of income and living standard, the population growth and urbanization, the relocation of industrial enterprises and developments of TVES, drive the city grows. When they are taking place, three main types of construction, which comprise the main part of urban expansion, were carried out, housing construction, industrial site construction and commercial area construction and agglomeration. The third development usually comes later after the first two developments complete or almost complete.

Since Zhongshan is a historical city, which has a very density population in the old city center, the new constructions mainly taken place in the fringe urban areas. When choosing new residential or industrial construction sites, the trade off between land value and commuting cost has to be taken into account. Ordinarily, the city center has the highest land value and highest population density and better utility services. According to the bid-rent theory, the new development will happen as near as possible to the city center while at the same time try to minimize the land cost which controversially lead to a trend of finding new sites far way from the urban center.

Closely linked with bid-rent theory, which consider the straight distance to the city centre, the transportation network are extremely important in the urban development. The investment in freeways and other transportation infrastructures makes travel faster and more convenient, thus reducing the cost of commuting and freight transport, consumers can enjoy cheap housing in the suburbs while paying less of a commuting-cost penalty and the enterprisers can have good transport capacity with low land costs. As a result, suburban locations look increasingly attractive as commuting and transport costs fall, which spurs suburbanization and leads to spatial growth of the city.

Another aspect is that the new developments are more possible to happen nearby the already built-up areas. Like the character of urban centre, built-up areas usually have better utility services and homogenise form of residential or industrial area are more attractive to the same type of development because it is possible that the special requirement for that type of development has already be met in that area. For instance, a built-up industrial area has convenience storage area and suitable electricity and water supply. So in this study, the neighbourhood externalities are assumed equally as important.

There are some constraints on the urban spatial expansion. The new development will choose a flat area rather than a steep slope area due to the excavation and fill costs. The local government has some policies that protect the water body and conserved green areas. It is not allowed to transform the current water body or conserved area into any other land uses.

The last but most important constraint is posed by the local authority that there is only a certain amount of land is available for a certain year according to the master plan and current situation. Any new development has to get permission from the municipality.

From the viewpoints of spatial scale, which indicates the distance one factor can affect, the following factors have to be considered:

- Local factors:

  Neighborhood (the total number of developed cells), in another word, the transition from non-urban to urban was much affected by the adjacent area. If the vicinity of a cell is developed area, this cell has much higher potential to transit to urban form if it was non-urban before.
- Regional factors:

  Natural constrains (terrain and water body) and road network.
- Global factors:

  Policies (master planning), this part controls the total space available for development, the developing direction, and the different developing speed for different area.

We can also categorize the factors in such a way that they are in two groups: non-constrained factors and constrained factors. The constrained factors indicate those limitations posed on the urban development such as water body is not allowed to develop into urban use, high slope terrain is not suitable for urban use, reserved area and public green area is not allowed to transfer into urban, etc.

From this point of view, factors can be organised in such a way:

Non-constrained factors:

- Neighbourhood

  Urban form itself has a very heavy influence on the urban development. At a small scale, a city tends to grow in the areas nearby the developed urban area. Here introduces the CA principle that the neighborhood affects the further development.

- Transportation network

  Transportation network have proved to be the key actor in urban evolution process. It heavily impacts the urban form and land price and even the location of city center and sub-center. While the relationship between road network and urban form is far from clear. It is an interaction process. High accessibility area causes intensive developing activities and these activities result in high-density areas with insufficient accessibilities. Subsequently, the high-density areas are possibly causing the upgrading of old roads.

- Urban centre and sub-urban centre (regional factors of urban form)

  From the traditional urban theory, land development will take place around the city centers first then further. They provide more energy, better facility services, more working and business opportunities, etc.

Constrained factors:

- Natural constraints

  Terrain and water body set the barrier on urban expansion. Slope and geographical conditions limit the developing interesting and costs.

- Policy constraints: Land supply and government management

  The municipality control the land supply and the developing direction of the city somehow. Any construction has to get the permission of development from the local government before its initiation. The local government thus impose their intents and limits on the urban development in terms of land supply and developing permission. It is a typically top-down process.

Furthermore, the factors are separated into compensatory and non-compensatory groups. Table 5 shows the summary of the above discussion.

Table 5Factors that affecting urban form

Factors	Local	Regional	Global	Compensatory
Neighbourhood function	Neighbourhood			Yes
Urban form		Centres Distance decay function		Yes
Transportation		Accessibility		Yes
Natural constrains	Reserved area/public			No

	green area			
Master plans			Developing direction Defined as resistance coefficient(r)	No
Policy constraints			Government control of land consumption	Not in this category

4.3. Conceptual CA model to simulate urban growth

After the affecting factors have been identified, a conceptual model framework has to be developed to incorporate these factors into a CA model, which will be implemented later.

4.3.1. The general principle of development of a CA model

As mentioned in chapter 2, the classical CA model definition has to be relaxed a lot during the development of a CA model that can mimic the real city development. Furthermore, there is no one common recognized best way to of building CA or MAS (multi-agent system) model structures. Different architectures have merits depending on the purpose of the simulation (Gilbert and Terna, 1999). One has to choose the model structure that can maximize the similarity of the model result and the reality according to the purpose and data availability. Moreover, the selected model structure should able to explain processes rather than be a black box and it should be able to use minimum number of parameters.

Since CA model is a micro-scale model to model the emergence phenomena from the smallest geographic units. There are three key building blocks from which CA model is built, Cells (individuals), neighbourhood and rules (transition rules)(Roger et al., 2002). The cell has to carry many attributes which will be used during the model running, but the increasing of attributes number in turn decrease the model performance in terms of memory consumption. How many states should one cell represent is also an issue of the focus. This has to be done according to the data and modelling purpose. For instance, when modelling the land use change, one has to define the number of states at least equal to the number of land use types. The same consideration has to be applied to the definition of neighbourhood. For instance, to model the residential dynamics will use a smaller neighbourhood definition compare to the neighbourhood size used by a model that is to model the industrial relocation process due to the different consideration of neighbourhood. An industrial site selection will consider a large area while a residential construction will consider a relatively small area. However, the variety in size and shape will heavily influence the model performance. Therefore, it is better to use a small neighbourhood definition when it is acceptable.

The definition of transition rule is the most crucial part. In most of the cases, the transition rules are set based on the understanding of the study phenomena by the modellers. Thus, the transition rules should be straightforward, simple and understandable because one reason of use CA model to simulate complex world is that CA model can generate great complex phenomena based on fairly simple transition rules.

4.3.2. Basic assumptions

Before coming to the CA model structure for simulating urban growth, the basic assumptions should be set up. They will act as the prerequisites to the development of the CA urban model.

Based on the literature and field observation, three assumptions are used in this CA model:

- The city will only grow, no reverse anti-urbanization process will happen in this model.
- Only two states can one cell possesses, urban and non-urban. No other intermediate states will be considered in this mode.
- The urban developments speed remains the same within each period.

4.3.3. Configuration of model structure and elements

To model the urban growth process in CA model, the very first step is to discrete the continuous urban space in to discrete space represented by lattice. Then the consideration is focused on the cells, which together comprise the lattice, urban space. Each cell in the lattice, urban space, evolves through time changing its state according to the affecting factors identified in section 4.2.2 such as neighbourhood, transportation, natural constraints and policies, etc.

Since we are focusing on the urban growth process, namely the urban and non-urban transition, only two states are used in this CA model: urban and non-urban. In this CA model, the urban development only takes place in the non-urban cells. A certain amount of land is available for development from non-urban to urban for each year within the model duration. The non-urban cell calculates its developing potential value each year based on the MCE method, which combine all the factors identified. Each year, The root agent rank the potential values of all the non-urban cells into an descend order and determine a threshold above which the non-urban cell will develop into urban cell if its developing potential values is higher.

Hence, the basic CA model can be configured as following:

- The state of a cell only has two states: urban and non-urban.
- Each cell has a developing potential value that is determined by the combination of all the affecting factors using MCE method.
- An overall “agent” called “root agent” controls the land consumption and time synchronization.
- The model will convert those cells that have developing potential values higher then a threshold for each step.
- The threshold is determined by the overall “agent” using a rank process which ranks all the cells according their developing potential values in an descend order and then select the top most ones that exactly fit the land supply by the agent in that year.
- The model time step is set to one year, which means one model step equals one year in real world.

The term “agent” is borrowed from multi agent system (MAS) terminology. In MAS, an agent is defined as an autonomous entity or object which acts independently of each other as well as its surrounding environment. It can sense the environment and act on it. An agent has its own goal or agenda to pursuit overtime. ((Batty et al., 2002)). The concept “agent” covers a wide variety of behaving objects from human to animals, etc. Agents can be classified in two classes: “reactive agent” and “cognitive agent”. The basic distinction between these two is that the reactive agents behave entirely to its environment and other agents while the cognitive agents also behave according to their own plans or goals. In this model, the root agent will play as a reactive agent that its one task is to collect the developing potential values of all the cells and then rank them to determine the developing threshold.

As mentioned in section 4.2.2, the affecting factors in urban development were classified in two groups: non-compensatory factor and compensatory factors. Thus, the MCE method can be introduce in the model to represent the evaluation process of these factors. The developing potential value can be defined for each cell as:

$$\mathbf{P = C} * \sum W_i * E_i$$

Where P denotes the potential value of current cell, C denotes the non-compensatory factor effect that in this case is conserved area, W_i denotes the weight for compensatory factor *i* , E_i denotes the effect of compensatory factor *i*.

The conceptual design is given in Figure 7.

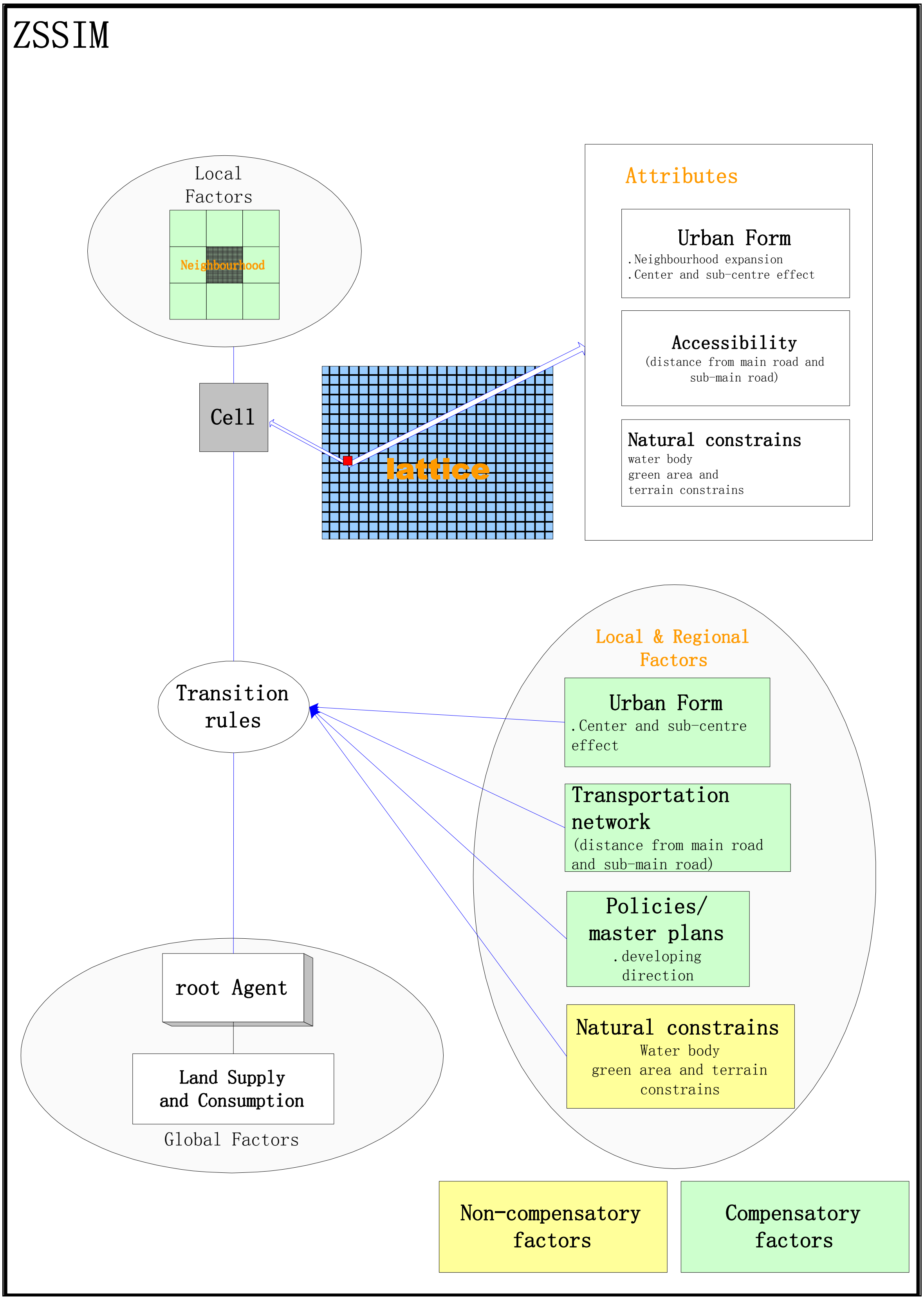


Figure 7 The conceptual model structure of "ZSSIM"

Figure 7 shows the conceptual configuration of the CA model, which is to be implemented to simulate the urban growth in Zhongshan. The model is called "ZSSIM". The whole model consists of four components: the root model "ZSSIM" (including the root agent), the Lattice that can be populated by the cells, cells and transition rules.

The root model "ZSSIM" acts as a container to put the other model components in it. Another function of root model is that it is defined as a root agent to take care of the whole model performance such as how fast the city grows, when the new roads should take place, etc. This root agent controls the land supply and consumption in the model world based on the historical data. It also manages all the control parameters used in this model.

A lattice represents the area of the study city, Zhongshan. It consists of regular grid cells. The total number of cells in the model is fixed through time. Each cell contains only two mutual exclusive states: Urban and non-urban. This indicates that each cell can have only one state, urban or non-urban, at time t. In this way, urban growth is conceived of as a transition from one state to another, namely from non-urban to urban. Considering the model performance in terms of time, the lattice has to be defined as small as possible, which means using large cell sizes.

Beside the state, a potential value of development is also attached to each cell. It will be used to represent the potential of a cell to be transferred into urban from non-urban. Other attributes are also attached to the cell such as accessibility, distance from the centre and sub centre, terrain limitation and policy gradient, etc.

The red spot in the lattice represent an example of a cell that carries its attributes. (Figure 6) Its state in next model step will be determined by the transition rules according to the affecting factors. As shown in Figure 7 , the affecting factors are distinguished by different colours, the green colour denotes the compensatory factors and the yellow one is non-compensatory factor.

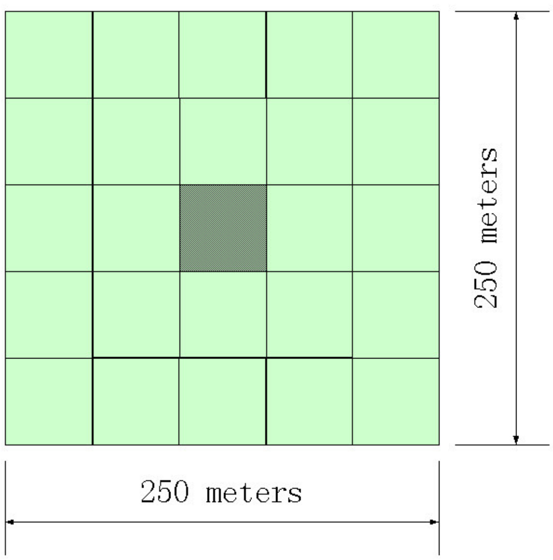


Figure 8 Neighbourhood definition

The cell considers its adjacent cells as its neighbours. The neighbourhood is defined as Moore neighbourhood. Figure 8 shows an example of a 5 by 5 Moore neighbourhood. The size of the neighbourhood will be decided after the sensitivity analysis since it will influence the model performance heavily. The large neighbourhood definition will slow down the CA model dramatically because it has to be calculated for each cell in each model iteration.

The transition rules are the most crucial part of CA model. Cells can only evolve and change their state over time through transition rules. Two transition rules are defined in ZSSIM mode:

- CalculatePotential rule
- Update rule

The CalculatePotential rule utilizes a MCE (multi-criteria evaluation) method to determine the current developing potential for each cell and inform the value to root agent. . Since a MCE approach is used in transition rule, the representation of affecting factors in the next section have to be standardized and ranged from 0 to 1. (Value range 0 to 1 is just for convenience, we could also use a data range from 0 to 100). This is because the effects of the influencing factors are measured and calculated in different units. To make them comparable, they need to be standardized. Moreover, standardization allows the expression of preference regarding the raw data. For instance, when considering a slope map, for some purposes (skiing) a steep slope is preferable while for others (agriculture) steep slope are not suitable. Standardization allows the different interpretation of the same data where a value of 0 indicates the least appreciated raw data value and a value of 1 the most appreciated raw data value.

The Update rule is quite straightforward. It just converts those cells which have higher developing potential values then the threshold values gotten from the root agent.

The detailed definition of transition rules is introduced the transition rules in detail. (See 41)

4.3.4. The program flow

When the model start, the root agent sets all the parameters such as neighbourhood size, neighbourhood geometry, annual land demands, weight system for transition rules, weight system for factor effect calculation, etc. After this model environment is set up, all the cells in the lattice start iteration.

The transition rules will be applied to each of the cells. In one model step, two transition rules were applied to a cell. One is to calculate the developing potential value for this cell and send this potential value to the root agent, which is the model itself, and the root agent put this potential value in an array to store all the potential values of all the cells in the lattice. The root agent will rank all the potential values into an descend order after all the cells have notified their values. Then according the annual land consumption which comes from the real data analysis and has already been translated into certain amount of cells which will be developed into urban, the agent can decide a threshold to be used by the next transition rule, update rule. The threshold indicates that the number of the cells who have higher potential values than the threshold is exactly the same number as the land consumption amount. For instance, if there are 500 cells available in one year for urban development, and there are 3000 non-urban cells have potential values scattered from above 0 to 1, after ranking these 3000 cells by their developing potential values into descend order, the top most 300 cell are chosen to be developed into urban cells. The developing potential value of the 300th cell is the threshold that will be used in Update rule.

Then another inner iteration starts, applying Update rule. When Update rule is applied to each cell, it compares its potential value with the threshold and if its potential value is larger than the threshold and it is previously not a urban cell, it will be transited into a urban cell. After all the cells applied this update rule, the model starts another iteration, which means a new time step. The all cells calculate their developing potential values again and those qualified cells transit. (see Figure 9)

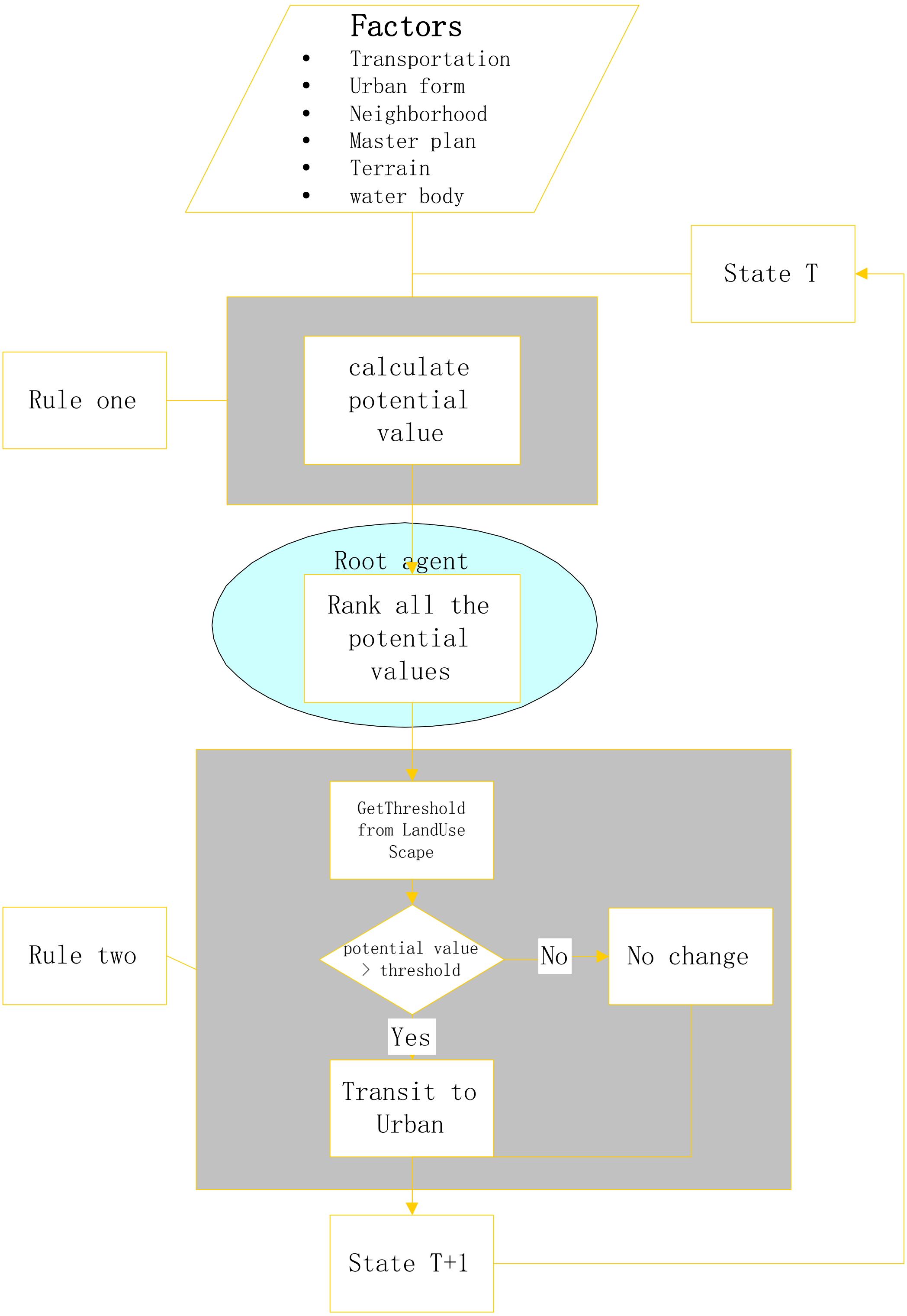


Figure 9 Program flow

Figure 9 shows the process on one cell in one model step. The Grey colour box denotes the transition rules and light blue ellipse denotes the root agent. Rule one is CalculatePotential rule and rule two is Update rule.

4.4. The representation of factors in CA model

The effects of those influencing factors identified in section 4.2.2 are used in the Calculatepotential rule evaluated by a MCE function. They are stored as the attributes in cells. All the effects of these factors will be standardized to a numerical score, whose scale is 0 to 1, so that the effects of them are comparable to each other. The measurement and standardization processes of these factors are describe in following sections.

4.4.1. Neighbourhood effect

Urban development usually takes place first in the adjacent area to the developed urban area. That is where CA model can make sense. Considering only the neighbourhood function, the potential value can be defined as:

$N = f_{xy}(q, Num)$

Where $f_{xy}(q, Num)$ is a function of the neighbourhood for cell in location (x,y), q is the developed cells in the neighbourhood and N is the number of the neighbours. The simplest way to define this is to use the q as the amount of developed cells in the neighbourhood and use Num as the total number of its adjacent cell in its neighbourhood. In this sense, f is define as

$f_{xy} = q / Num$

Where q is the number of developed cells in the neighbourhood and Num is the total cell number of the neighbourhood. More complex definition can introduce the direction control into the neighbourhood function using weight system in neighbourhood.

4.4.2. Road network effect

The transportation network influence is usually calculated in terms of accessibility. To make the calculation simple and effective, only two types of roads are considered, main roads and secondary roads. Based on the day-to-day observation, we could see that the road network influences the urban form very much. A weight system has to be applied to the different road classes.

Another consideration is that the effect of road network is always being considered to be a distance decay function. The use of a power distance-decay function in computing interactions is borrowed from Newton's famous law of gravitation. Consequently, these types of techniques are sometimes called gravity modelling. Two kinds of decay functions are commonly used: power exponent decay and exponential decay. (ESRI)

In terms of power exponent decay function, the accessibility is defined as:

$$T_i = \sum_{j=1}^{n} W_j d_{ij}^{-\beta}$$

Where T_i is the accessibility of cell i, W_j is the weight for road class j, d_{ij} is the distance from cell i to the nearest road class j, β is the exponent.

In the case of the exponential decay function, the equation used is:

$$T_i = \sum_{j=1}^{n} W_j e^{-\beta d_{ij}}$$

Where T_i is the accessibility of cell i, W_j is the weight for road class j, d_{ij} is the distance from cell i to the nearest road class j, β is the exponent.

These two equations have the same function with different decline rate decay curves as shown in Figure 10.

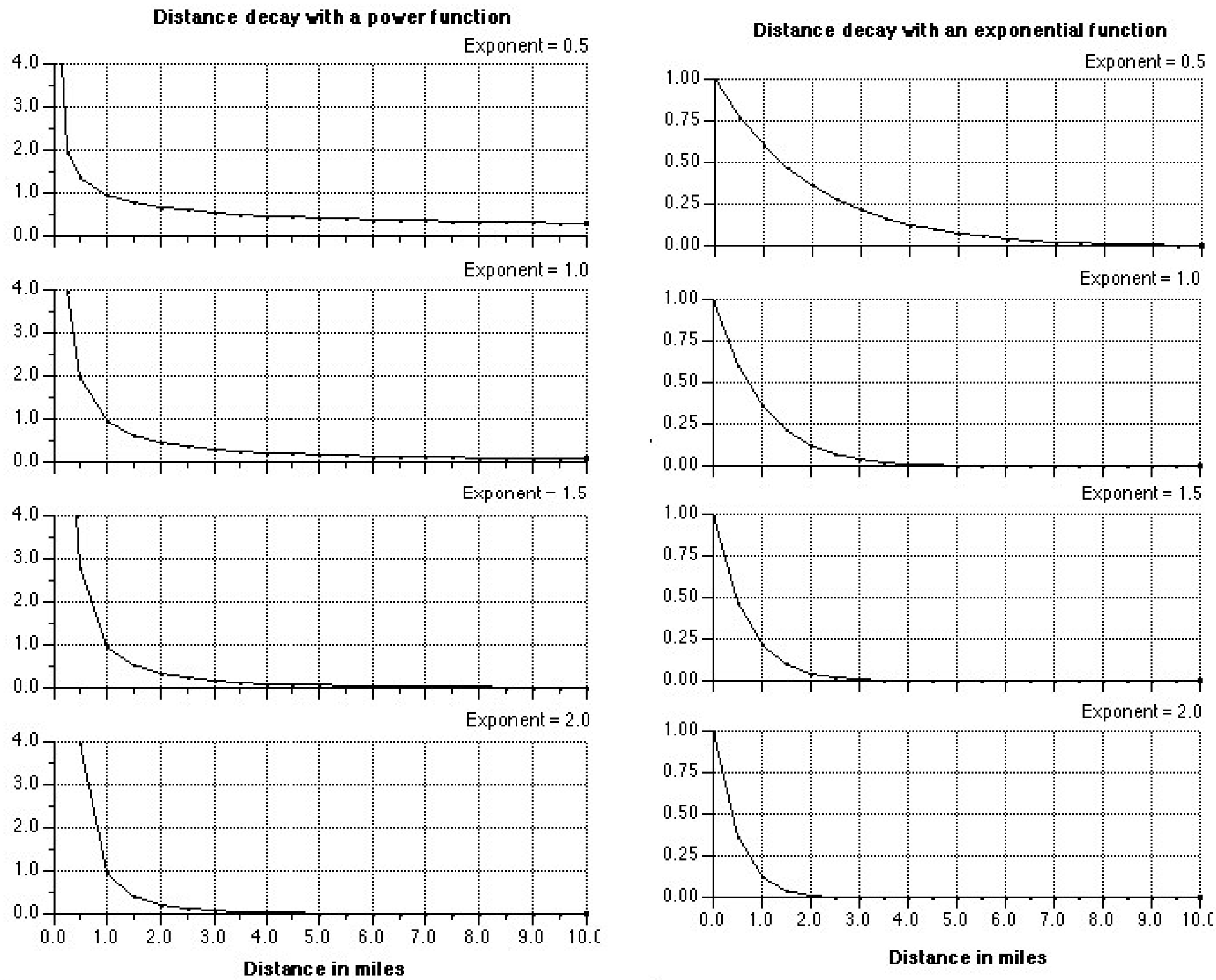


Figure 10 Distance decay functions (ESRI)

The main concern on road network is about accessibility which has already been discussed quite a lot if not thoroughly. The common concept on accessibility is that the effect of road, which influences the location choosing process for residential area, is a function of power exponential decay or exponential decay. (ESRI)

As the transition rule is a multi criteria evaluation function, all the effect of different factors should be standardized into a score ranged from 0 to 1. It is straightforward to choose the exponential decay function that already normalized the results.

For choosing the decay exponent number, it is mainly depends on the mobility of the habitants in the city. High mobility leads to lower decay exponents with slower decay rate, which means that people with high mobility can easily travel far compare to those who have low mobility. (See Figure 10)

In this research, the exponential decay function is used due to its nature of automatic normalization.

4.4.3. Urban form effect

According to the classical bid-rent land value theory, urban developments tend to take place as near as possible to the city centre. Thus it can also be assume as a distance decay function. The gravity model was borrowed again here to reflect the city centres effects in terms of attractiveness.

$$A_i = \sum_{j=1}^{n} W_j e^{-\beta d_{ij}}$$

Where A_i is the attractiveness of cell i, W_j is the weight for city centre j, d_{ij} is the distance from cell i to centre j, β is the exponent.

The urban centres are defined as points and the weight of each urban centre is set by visual interpretation of the urban development maps such as land use maps in different years. Following is an example of the influences of two urban centres.

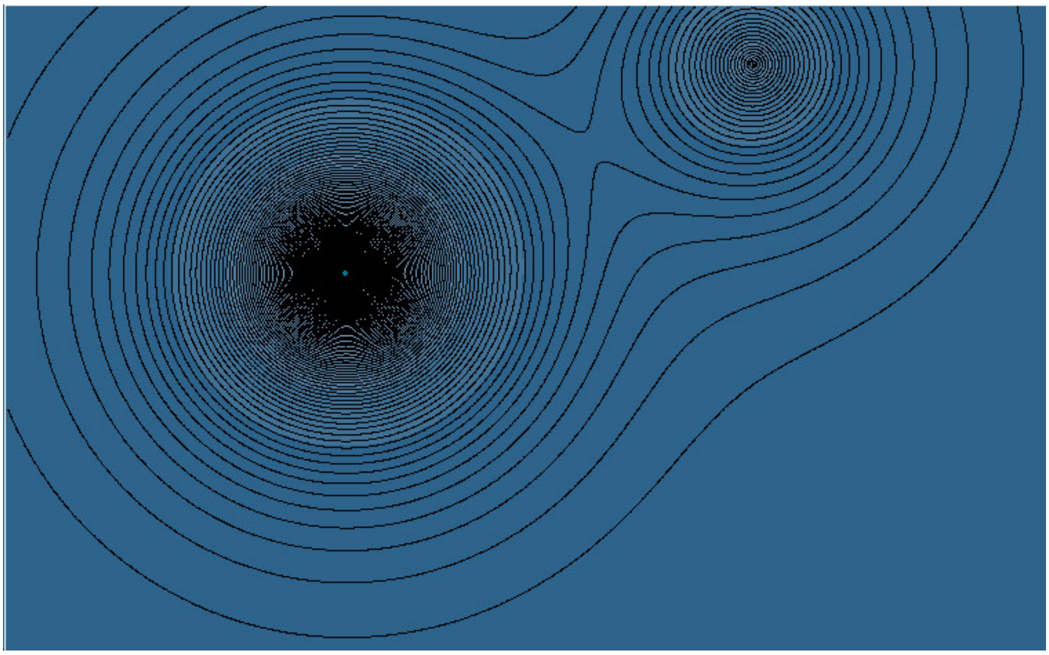


Figure 11 Attractiveness contour lines

Figure 11 shows the attractiveness of two urban centres. The contour lines reveal that the near to the city centre, the higher the attractiveness values. Here, weight for west urban centre is set to 0.7 and weight for east urban centre is set to 0.3. Decay exponential number is 0.5.

4.4.4. Terrain constraint

Ordinarily, city will grow where there are flatten areas. People tend to live in flat area rather than a hilly terrain. In this case, we assume that the slope will constrain the urban form mainly depends on the slope. Terrain constraints will be represented as a slope map in percentage. In a percentage slop map, cells with value 0 represent the most flatten areas while cells with value 100 represent the 45 degree areas.

Thus, the effect of terrain constraint can be formulated into:
$T = 1 - (Slop/100) \quad (slop < 100)$
Where T is the effect of terrain constraint, Slop is the slop of the current cell. Limited the slop below 100 indicates that we conceive that a cell with slop more than 45 degree is entirely unsuitable for urban development.

4.4.5. Water body and conserved area

Water body and reserved areas are not allowed to be developed into urban land use whatever the other conditions were. This factor is the only non-compensatory factor in this CA model. Only two states are defined for this factor: 0 denotes the current cell is conserved which is not allowed to be developed into urban cell and 0 denotes the current cell is outside the conserved area and no limitation on its development.

.

4.4.6. Master plan

Master plan shows the particular direction of developing that the local government interests in. it is quite difficult to represent this factor. While in city Zhongshan, the master plan clearly shows the municipality intends to develop the urban area towards the new city centre which is located right to the east. In this sense, in this research, the master plan is assumed to only guide the urban development direction. The effect of master plan is represented by a linear gradient function as:

$M = x / length$

Where M is the effect of master planning, x is the horizontal coordinate in X axis, it starts from 0 and ends at length of the lattice, length are the length of the lattice.

For example, one cell at the west edge which has a x coordinates value 0 will get a master plan effect value 0, and one cell at the east edge of the lattice will get a value 1 since its x coordinate equals to the length.

4.5. The transition rules

The transition rules have already been discussed in section 4.3.3, there are two transition rules which control the cell states transition: CalculatePotential rule and Update rules. In this section, the detailed formulations of these two rules are given. In each model step t, CalculatePotential rule calculates the developing potential value for each cell and informs the root agent the result, Update rule gets the threshold value from root agent and uses it to decide which cell should be developed into urban cell.

4.5.1. CalculatePotential rule

As described in section 4.3.4, cells evolve by time steps. In each time step t to *t*+1, the cell in location (x,y) changes its state according to the states of its neighbourhood and considering the other factors or constraints. To determine which cell should be converted into urban land use, an appraisal to each cell has to be carried out and a potential of transition to urban land use was calculated according to the natural condition and its neighbourhood situation. This is done by applying transition rule Calculate-Potential rule. In CalculatePotential rule, a multi criteria evaluation approach has been used to appraise the potential of conversion. Following is the equation used in CalculatePotential rule:

$$\mathbf{P = C} * \sum W_i * E_i$$

Where P denotes the potential value of current cell, C denotes the non-compensatory factor effect that in this case is conserved area, W_i denotes the weight for compensatory factor *i* , E_i denotes the effect of compensatory factor *i*.

After defining all the factor effects representation, now the equation can be expanded as:

P = C * (W1 * N + W2 *R + W3 *A + W4 * T + W5 * M)

Where P is the developing potential value, C is the conservation limitation, N is the effect of neighbourhood, R is the effect of Road network, A is the attractiveness of urban form, T is the constraint of terrain, M is the effect of policy and master plan, W1 to W5 is the weight system applied to the factor effects.

C is a binary variable, 0 or 1. Value 0 denotes conserved area which is not allowed to develop and value 1 denotes areas outside conserved area and is able to be developed into urban cell. Variables N, R, A, T and M range from 0 to 1. Weights W1 to W5 also range from 0 to 1 and the sum of all weight equals to 1.
W1 + W2 + W3 + W4 + W5 = 1

Thus, through adjusting the weight system, the model can mimic the different effect of different factor influences.

Another function of this transition rule is to report the result developing potential value of current cell to the root agent. After calculating the developing potential value, CalculatePotential rule sends this value to the root agent. The root agent stores the value in its array so that all the values can be ranked into a descend order after it gets all the developing potential values for all the cells.

4.5.2. Update rule

After all the cells have applied transition rule CalculatePotential and the root agent has set the transition threshold, the Update rule is fired to convert those qualified cells into urban cells. This rule transits the current cell's state to urban land use according to the threshold gotten from top-level agent. If the current cell has a higher potential value than the threshold and it is a non-urban land cell, it will be transferred into urban land use.

4.6. Modelling the two periods of urban development

The city Zhongshan has experienced a dramatic change in developing direction as well as developing speed during the last two decades. This mainly subjects to the economic boom and policy guidance change. Whilst it is difficult to represent this kind of incidental events since CA is supposed to be a close self organized system.

There are two solutions to simulate these two period developments: one is to implement them separately and make two separated models, another is to change the model control parameters as well as the input data during the model year 1992 in which the change had taken place in the real city.

The affecting factors can be examined to find what's the different between these two periods in terms of input data. Neighbourhood effect is always a consideration of the current situation of the nearby cells, so we may use the same neighbourhood definition in the two periods. The road network effect is quite different in the two periods. Several new main roads were constructed and started affecting the urban form in the second period, from 1992 to 1998. Urban form effect is also different in the two periods. Before 1992, the new urban centre, Zhongshangang centre, has very less influence on the urban development. After 1992, the local government intended to build the new urban centre as a new technology-developing zone and thus came the massive investments and construction which formed a new urban centre. The natural condition such as terrain, water body and conserved areas remained almost the same in the last two decades.

From the description above, we can only consider the rapid growth of the new urban centre and the construction of new road network in the model since the other factors remain the same in the two periods. Thus, the two periods of urban development will be represented in the same model by changing the control parameters and the input data for road network. In ZSSIM model, the new road network data will be imported in 1992 and the new urban centre will increase its influence by increasing the weight in 1992. The land consumption amount will also be changed according the historical data.

5. Model implementation and calibration

In this chapter the implementation aspects of ZSSIM model, the CA urban growth simulation model for Zhongshan city is presented. During the conceptual data modeling stage, implementation details and system requirements were not considered. These topics are considered in the context of the implementation of the CA model. The calculations of the scores of, as those factors well as detailed definition of model elements in terms of software implementation, which are used to realize the model, are explained in this chapter. The developed conceptual model is implemented and the model is calibrated and evaluated.

Starting from selecting the development platform, this chapter covers all the operational part of this research till the evaluation of the implemented CA model.

5.1. Implement platform

Before we move to the step of choosing development platform, we have to decide which kind of relationship this CA model should adopt. Two kinds of CA-GIS coupling approaches could be used to implement ZSSIM model. Each of them has their own advantages and drawbacks: close-coupled approach or loose-coupled approach.

In a close-coupled environment, CA concept is incorporated into the raster GIS system due to their consistency in the spatial representations. Usually, this kind of development uses the script language comes with the raster GIS system. The limitation of this approach is obvious, the script languages are usually not efficient enough to carry on a computation intensive task, the GIS is design to be an efficient data acquisition, analysis and management tool rather than a dynamic modelling environment, the script language usually are not as flexible as those general programming languages such as C or Java. It will take more efforts to deal with the model structure issues when the CA model grows sophisticates.

In a loose-coupled CA-GIS environment, CA model is implemented separately from the GIS. Thus, the model developer can use the most suitable programming languages in terms of efficiency and take the advantages from their former experiences in programming as well the advantages of Object-oriented programming concept. Moreover, the model is easier to be optimized in terms of performance. The drawback is that extra programming effort has to be taken to facilitate the data exchanging mechanism.

In this research, the efficiency or speed of CA model is more crucial because the lattice is quite big in this simulation effort. The loose coupling method is chosen to implement the CA model and the CA model exchanges data with GIS via TIFF files. (see Figure 12)

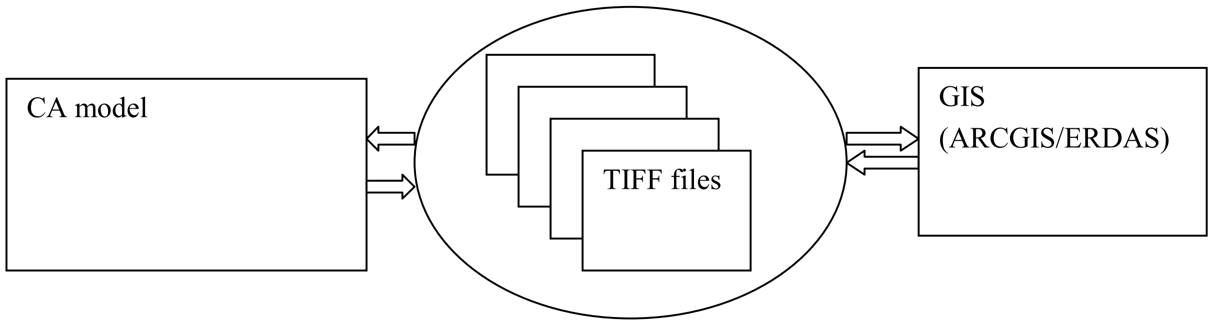


Figure 12 the relationship between GIS and CA urban model

5.1.1. Guideline for choosing the implementing platform

There exist many CA model tools and software and source codes. Basically there are three options when implementing a conceptual CA model:

- Using an existing CA model,
- Developing a new CA model from scratch
- Using a development framework to construct a new one

The concept of CA is quite simple and it is not very difficult for a experience programmer to build it from scratch. Also some general purpose CA developing platforms are available which claim that they are easy to use. However, the CA systems available on the market or Internet are not specifically tailored for general-purpose modeling though some of them announce that. As Parker (Parker, 2001) argued, three reasons make the majority of model developers in CA and Multi Agent System (MAS) area develop their own model rather than using a existing one. First of all, this kind of modeling efforts seems quite simple and straightforward. It looks simple to construct a CA model just from scratch. Secondly, agent based modeling frameworks typically appear daunting and the benefit of such tools does not seem to warrant the cost of learning it. Thirdly, software developers and researchers are culturally very sensitive to loss of control and are not comfortable with subjecting important parts of models to a "black box" implementation. To decide which approach should be taken, one has to balance the ease of development and loose of control of the model details.

Further more, the potential of extending the CA model into more complex structure model such as MAS/CA hybrid model is another consideration. It is quite straightforward to think about to extend the CA urban growth simulation model into a hybrid MAS/CA model with population and social economic data incorporated in. For example, in a hybrid model environment, we can used some road constructor agents which will moving around the city and monitoring the current accessibility, if some urban area have very low accessibility then they will construct new roads there. Also, we can use different stakeholder agents to represent the different roles in urban developments. For instance, resident agents will choose their resident sites and industrial agents will construct industrial sites according to their different requirements and affecting the city through their different behaviors. In this sense, when choosing the development platform, we have to consider the extension ability of the developing platform so that we can extend the model later easily.

Another perspective is that CA model is a computation intensive process. To simulate the urban growth phenomena, enormous computation power is needed. The calibration process is a time consuming work. Manually calibrating a CA model requires the model to be able to generate result fast so that the adjustment can be done easily. The developing platform has to use a relative efficient language in terms of performance.

In summary, an appreciated developing platform should have the following characteristics:

- Concise, elegant structure
  Makes it easy to understand and learn and be extended

- Efficiency in computation and development
  Mainly depends on the developing language. A platform developed in object-oriented language is more appreciated since the object oriented languages are thought to be suitable to represent the real world features while at the same time some of them are fast enough such as C++ and java languages.

- Open source
  The model developer could have the full control of all the model details if they intend to.

There are several available open-sources MAS/CA developing platforms such as Startlogo, MAS2, UrbanSim, Ascape, etc. could be used as the developing platform. Ascape is chosen as the developing platform for this research due to its flexibility and potential of extending the CA model to more complex structure and the fact that it is an open source developing environment.

5.1.2. Introduction of Ascape

Ascape is a collection of Java classes developed by Miles T. Parker at the Brookings Institute. Ascape is full modeling environment for development, analysis and distribution of Agent Based Models. "The design goals include abstraction and generalization of key agent modeling concepts, Ease of Use and configuration, best attainable performance and ease of deployment. It is written completely in Java in order to provide maximum deployment options, and to take advantage of Java's strong typing and idioms" (Parker, 2000).

The most important concept of Ascape is Scape, without understanding of what it is, it is not possible make use of the power of Ascape. Scapes are collection of agents and can be iterated across agents. In this sense, they are the containers which hold a group of agents. They know not only what they contain, but also know how to iterate across what they contain, detailed topology of what they contain. And they are also agents themselves which is rather confusing but very powerful. This will be explained later.

The Ascape design provides a number of key abstractions that facilitate the composition and exploration of agent based models. Ascape relies heavily on three abstractions that determine model structure and layout. These are:

- All scapes are themselves agents.
- Scape structure is hidden from agents.
- Behaviors occur across scapes, as 'rules'.

All scapes are themselves agents

Basically, everything is an Agent. Scapes are essentially collections of agents with plenty functionalities. They contain agents or other scapes and iterate across the agents or scapes contained by them. Anywhere that a model can use an Agent, it can use a Scape. Thus, scapes can always be composed of other scapes. This allows a very natural method for model composition. It also ensures that any behavior that occurs across an Ascape model occurs in a well-defined way. In Ascape, models are always composed of hierarchies of Scapes and agents.(Figure 13)The effect of this is that the model becomes strictly hierarchical and highly composable, where each system can impose any topology (form) on its constituent subsystem, and no "hidden" links exist which could create an undesired interaction. In every multi agent model developed in Ascape, there must be a root Scape to contain all the other Scapes. Since all Scapes (Agents) belong to the root Scape, all system behaviors (such as statistic collection, iteration rules, visualization) are managed systematically across the class hierarch. For instance, the root Scape has its own rules which will only be applied to the root Scape and its constituents Scapes have their own rules and the rules will only affect the Scape which the rules are attached in.

Scape structure is hidden from agents

This abstraction means that agents do not directly interact with the "space" they exist in. Agent behavior including searching and neighbor interactions is designed without respect to a particular topology. The agents use the information that the upper level Scapes, which contains them, provide them when this topological information is necessary. This character enables the Ascape model to be put in different topologies through a single line of code, the agents to be "space independent". Also, this abstraction makes sure that all the algorithms are placed in Scapes they belong to. This creates logical transparency and reduces the amount of code necessary, and keeps it at one place, making modifications highly effective.(Parker, 2000) For example, when using search neighbour function, an agent has no idea about the space it is living in, the space might be a periodic space or an aperiodic space, the Scape which contains this agent will be in charge of providing the search result to the agent.

Behaviours occurs across Scapes, as Rules.

In an Ascape model, Scape hierarchy is structurally determined. Behaviour occurs only across Scapes. There is no other high-level mechanism for executing behavior on agents. Therefore, the behavior of Agents is executed as rules collectively across a Scape. Moreover, if necessary, rules can even be applied at a single Agent. This gives the Ascape model great flexibility of carrying out behaviours.

It is very useful to look at the inheritance structure of agents and Scapes first before one could utilize this developing framework, Figure 13. Ascape models start with a root Scape which is usually a subclass of the class ScapeVector. This root Scape acts as a container to hold all the other parts of the model and holds assorted global variables needed by the model, which typically used to control the model.

Most often the root Scape will hold at least two objects, one ScapeGraph, and another ScapeVector. The ScapeGraph represent the “field” or environment for the agents. The ScapeVector in the root Scape holds agents who populated in the field.

When implement a CA model in this framework, the second component, the ScapeVector in the root scape will not exist. The automata are the cells themselves comprise the lattice, ScapeGraph.

Readers who interest in Ascape can get more information in Ascape documents which come together with the platform.

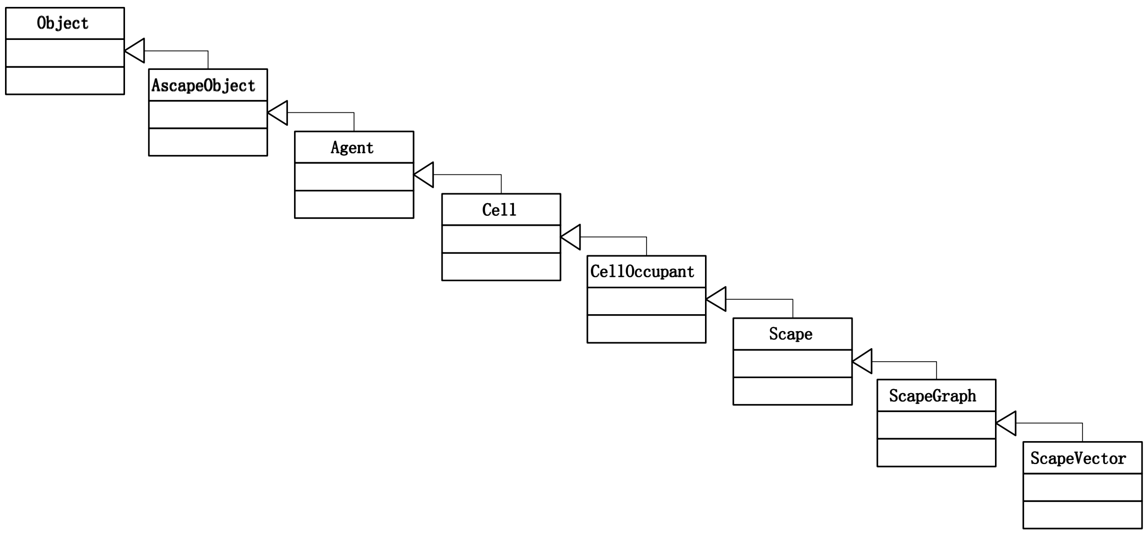


Figure 13 Ascape classes inheritance hierarchy

5.2. ZSSIM Model structure in Ascape

The conceptual model developed in chapter 4 is implemented in Ascape environment. Following Ascape class structure, the conceptual model components, the root agent, the lattice, the cell and the transition rules, are implemented in Ascape classes. The model follows a 3 level hierarchy structure: LandUse, Lattice and landCell. The LandUse, which is the ZSSIM model and root agent, at the top level, contains the Lattice which in turn contains the landCell, landCell is the implementation of concept cell in chapter 4. Transition rules are defined as variables in landCell class. The conceptual structure is presented in Figure 14.

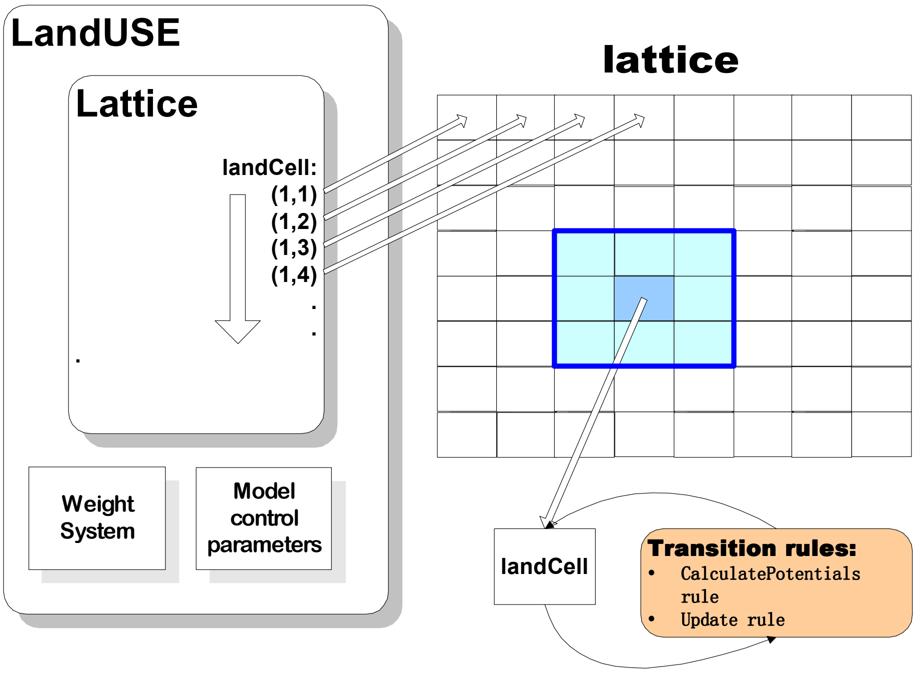


Figure 14 Model structure 1

Figure 14 shows the relationship of the LandUse, Lattice, landCell and transiton rules. The weight system is the weights for different affecting factors. It is contained in LandUse class. The LandUse class also contains the neighbourhood definition and other model control information such as dimensional information, model start and stop time, etc. Transition rules (in brown bulb) are implemented in landCell class and be repeatedly applied to each landCell during the model running.

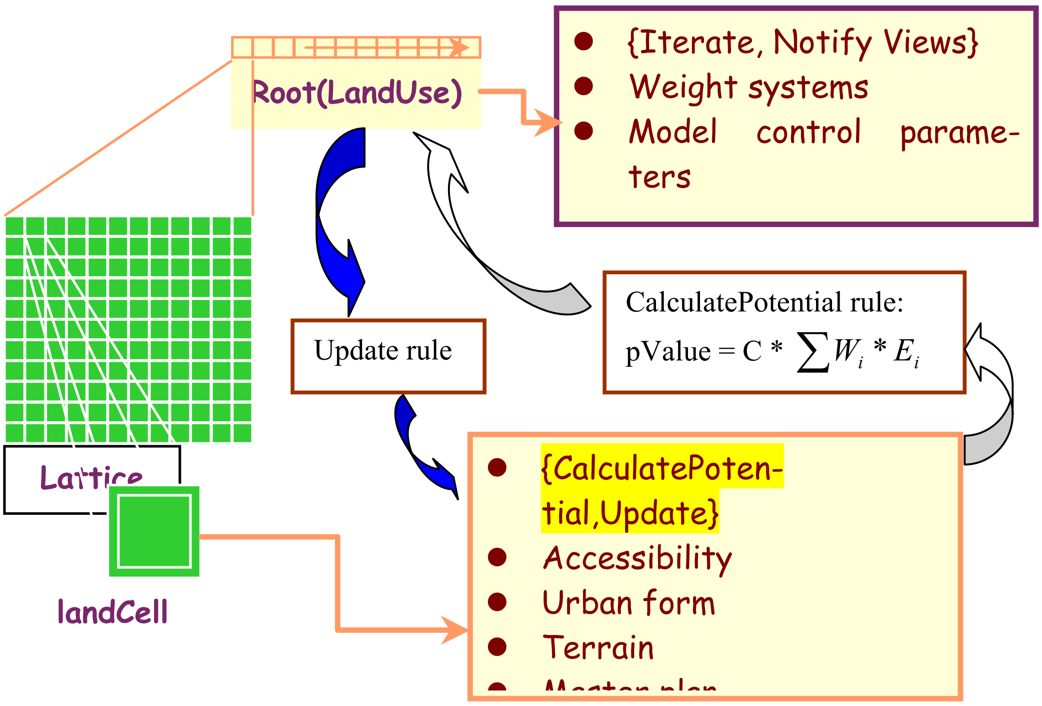


Figure 15 Model structure 2

Figure 15 presents another aspect of the implemented ZSSIM model. It shows the interaction between root agent and the landCell through transition rules. In each model time step, the landCell calculates its developing potential value and send it to root agent (the grey thick arrows). The root agent ranks the potential values for all the landCell and the landCell gets the threshold back through Update rule (blue thick arrows in Figure 15).

5.2.1. Model components

LandUse

The LandUse is inherited from ScapeVector Class and is the top-most layer of the model. It provides the basic definitions of the model such as model control parameters, etc. it also contains the lattice that landCells live in. The name LandUse comes from the first attempt to modelling the urban growth process considering different land use classes. After the two states model, urban and non-urban, has been decided, the name did not change.

Lattice

The Lattice represents the space of the study city, Zhongshan. The lattice is a container for the land-Cells. It provides spatial discretization to the model. The lattice consists of landCells and it defines the geometry within which the landCells will interact with its neighbours. Lattice is defined as a ScapGrpah class.

LandCell

The landCell, as the basic element of the behaviour owner, carries it own attributes and behaviours. The attributes provide the basic information about the current location such as distance to main road, distance to secondary road, etc. Here, the behaviours are exactly transition rules. LandCell is defined as Cell class.

Transition rules (behaviours)

Model is iterated in annual cycles. The two transition rules defined in Chapter 4, CalculatePotential rule and Update rule, are implemented as two variables in landCell class in Ascape.

The next section gives the detailed class definition in Ascape for each of the model components.

5.2.2. Classes defined

The model components are defined as Java classes under Ascape: LandUse, LandCell. Lattice is a variable defined in LandUse and transition rules are variables defined in landCell class. Figure 16 shows the class definition in UML diagram.

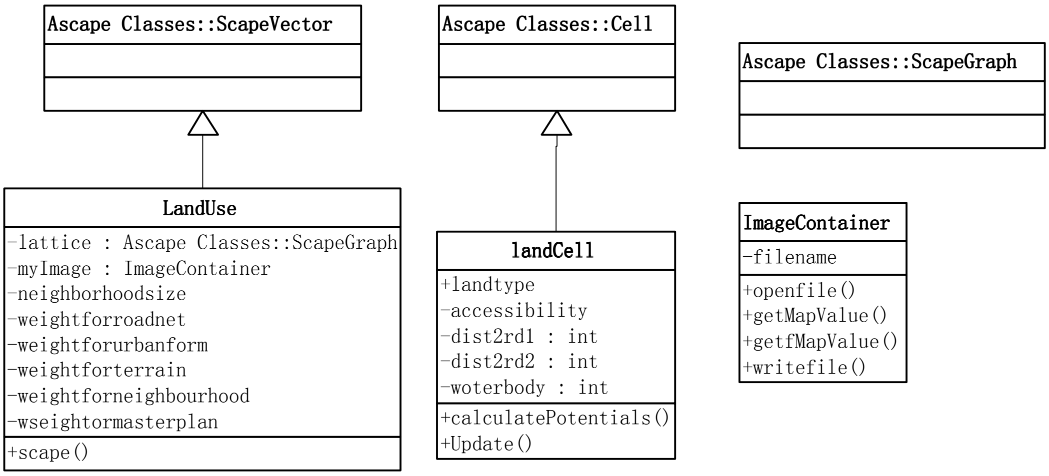


Figure 16 UML diagram of class definition

Class LandUse contains the weight system used by transition rules. All the weights are ranged from 0 to 1 and their sum should be 1.

Class landCell carries all the information about the current cell (location). It has two states(landtype): urban and non-urban which are represented by 1 and 0 respectively. Other attributes are:

Table 6 Attributes in landCell

Dist2mainroad	meters
Dist2secondaryroad	meters
Dist2mainCenter	meters
Dist2subcenter	meters
Waterbody	1 (non water and non conserved areas) or 0 (water body or conserved areas)
DEM	Slope in percentage

Most of the attributes in Table 6are self-explanatory. Two of them need to be explained: waterboday and DEM. The waterbody attribute represents the waterbody and also conserved areas. To use the MCE transition rule to calculate the developing potential value, the value is set to 0 where the cell is in water body or conserved areas which indicate no urban development will be allowed in this cell. Value 1 indicates a cell outside the water body and conserved areas and thus allowed to be developed into urban cell. Attribute DEM actually represents the slop map of the study city. The slop map is calculated in percentage.

Class ImageContainer is used to facilitate the data exchange with GIS and animation software. It contains the functionality of read and writes TIFF and GIF image formats.

For detail information, see Appendix 4. Source code. Usually the programming code will be translated to pseudo code when sharing the ideas. The purpose for that is to make the code easy to be understood. Java is a quite common used programming language, so in my appendix, I use the Java source code instead of pseudo code.

5.3. Parameter settings

All the model control parameters are located in LandUse class. They are:

- The dimensional information of lattice
- The weight system for prioritize the influence of factors on land use change
- The sub weight system for road network and urban form
- The overall land consumption control
- The model behaviour control

These model control parameters will be explained in the following paragraphs.

5.3.1. The dimensional information of the Lattice

The space (lattice) the built-up area of the study city, Zhongshan, is around 30 square kilometres, while the city form is a strap shape distributed along two main roads. To contain the whole city and the space for further development, a lattice of 30X18 square kilometres was chosen to represent the city and its surroundings. It includes all the built-up area and 7km space to east, 5km to west, 7 km to south and 3 km to the north. With the cell size in 50 meters, the lattice size is 606*376 cells (Figure 17).

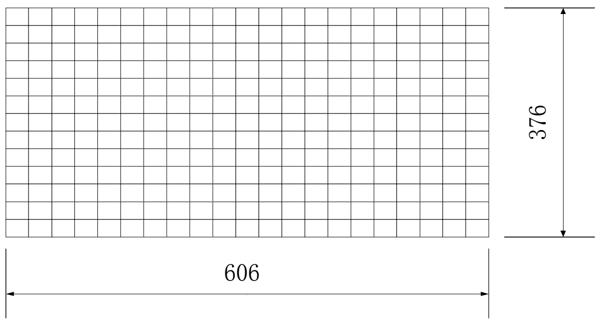


Figure 17 Lattice setting

5.3.2. The weight system for prioritising influence of factors on land use change

To use the MCE transition rule, the weight system among influencing factors can be manipulate during the model running. Different combination of weights for different factors could be tried out.

```
weight_Neighbourhood = 20,
weight_urbanForm = 20,
weight_Roadnet = 30,
weight_terrain = 20,
weigh_Masterplan = 10
```

The sum of these weights is 100.

Above is the initiate setting for the weight system. When using these weights in CalculatePotential rule, these weights are further divided by 100 to assure their range is from 0 to 1. Using integer numbers and setting the sum of them to 100 is due to a bug in Ascape that only integer numbers can be changed during the run time through Ascape interface. The weight system needs to be adjusted systematically to get the model resemble to the real city development and this will be done through the Ascape interface. That's the reason of using an integer number weight system.

5.3.3. The sub-weight system for road net and urban form

To balance the effects of two classes of road network, main roads and secondary roads, the road effect at a given location also needs a weight system to adjust. The same principle is applied to the urban form effect, namely the two city centres' influence.

For the road network, first the weight 0.7 was set to main road while 0.3 set to secondary road.

```
weight1ClassRoad = 0.7
weight2ClassRoad = 0.3
```

For the urban centres influences, the weighs for the two different urban centres have to be changed in the middle of the simulation. In 1980, the new developing zone centre was just a small fishing village and it had almost no influence on the urban form from 1980 to 1992. After 1992, the new centre developed rapidly and the development continuous till now. Thus, the weight for new urban centre is set to 0 at the start of the simulation from 1980.

```
 weightMainCenter = 1
 weightSubCenter = 0
```

In 1992, the weight for the new urban centre is changed to 0.5 which implies the equal influence of the two urban centres.

```
weightMainCenter = 0.5
weightSubCenter = 0.5
```

5.3.4. The overall land consumption control

This parameter controls the developing speed of the model. The developing speed has changed between the first period and second period. As described in chapter 3, the developing speed from 1992 to 1998 is much faster then from 1980 to 192. It increased almost twice in the second period. Based on the calculation in section 5.5.2, the land consumption is set to 459 cells per year in the first model period according to the statistic from the field data in 1980 and in the middle of the model running, in 1992 model year, it was changed to 946 cells per year, which consequently lead to a rapid model development.

5.3.5. The model behaviour control

Other parameters are used to control the model behaviour such as neighbourhood size, save data or not for each iteration, the model start period and stop period, etc. Neighbourhood size will be examined later in the sensitivity analysis to check its effect on the simulation result. Then the suitable size will be used for the following simulation. The model start period is set to 0 which denotes year 1980 and the stop period set to 18 which denotes year 1998. The ZSSIM model will stop at time step 18 under this setting.

5.4. trend break representation

CA models are usually conceived as close system, which has no data or information exchange with their external environment. The evolutions in CA model represent the self-organization process. Whilst for the top down process, which is a central controlled process, there is no mechanism in CA model to represent it well. External events such as the policy changes, the incidental natural disasters cannot be modeled in CA model without some central control mechanism being introduced into CA model.

When modeling urban phenomena, though we assume that the urban is in some extent a self-organized system, external events influence the urban developments in many perspectives. In the study city, Zhongshan, in year 1992, the urban development changed its direction as well as it speed due to the economic boom and policy changes. (See chapter 3) The economic boom leads to massive investments, land speculations and accordingly the policy leans to offer more land to the investors. To model this external influence in a CA model, a central control mechanism has to be used in this research. In section 4.3.3, a central control mechanism was introduced into the CA model, which is implemented by a root agent in the ZSSIM model. One task of the root agent is defined to take care of the external events and control the model to change its data and parameters in the middle of the simulation.

To reflex the significant change in developing speed as well as the effects of the new road network, in iteration 12 which means year 1992 in model city, the root agent will import the new road network which represent the road situation in year 1992 and replace the original 1980 road net work with it. The root agent will also change the weight for new urban centre from 0 to 0.5 and the old urban centre from 1 to 0.5 to reflex the suddenly increasing of the new urban centre and its influence. Finally, the root agent changes the land supply from 459 cells per year to 946 cells per year. (See section 5.5.2)

Designed in this way, the ZSSIM model is able to model the trend break in the two periods of urban developments for the study city.

5.5. Data process and analysis on urban growth

After the model structure and control mechanism has been set up and implemented, the data, which represent the effects of the affecting factors, have to be processed and imported into the CA model, ZSSIM.

5.5.1. Data preparation

To initiate the CA model, all the factors have to be import into the model, namely the urban/non-urban land use maps, transportation networks, terrain, water body and conserved areas. All the vector data of them are converted into ArcGIS personal geodatabase as feature sets.

- Urban non-urban land use maps derived from land use map
- Transportation networks were derived from transportation network map for year 1983 and 1999. They are edited according to the land use map because only transportation network map in 1998 was collected and road network in 1983 can be obtained from 1983 topographic map.

- Terrain information is considered to be a constant variable since in reality the terrain is also very stable. DEM was generated from elevation points and contour lines digitized from topographic map. Then it was covert to a slop map in percentage.
- Water body and conserved areas. Water body map was digitized from topographic map and conserved areas map was derived from land use map which notified that these areas are conserved as national parks and public green areas. Then the two maps were merged into one and reclassified into only two class category: 0, water body or conserved areas, means no development is allowed and 1 means this area is not constrained by this factor.

5.5.2. Urban area and land consumption

From the land use map, the actual urban area was calculated for each year of these three periods. Calculation was carried out in ARCGIS.

Table 7 Urban growth statistic data

<table>
<tr><td>Year</td><td>Total urban area (m2)</td><td>Increased by</td><td>Annual increasing rate(m2)</td><td>Annually developed cells</td></tr>
<tr><td>1980</td><td>5004644.562</td><td rowspan="2">13771098.097 (5508.4392388 cells)</td><td rowspan="2">1147591.508</td><td rowspan="2">459 cells</td></tr>
<tr><td>1992</td><td>18775742.659</td></tr>
<tr><td>1998</td><td>32972474.051</td><td>14196731.392 (5678.6925568 cells)</td><td>2366121.899</td><td>946 cells</td></tr>
</table>

In Table 7 Urban growth statistic data, the actual urban growth was translated into cells. The increasing of developing speed can be easily seen that the developing speed in second period is almost as twice fast as the first period. The number of annually developed cells is used by the root agent to control the developing speed in ZSSIM model.

5.6. Calibration and evaluation

Calibration is used to estimate a model’s parameters that provide the best fit to an observed set of data— the ‘goodness of fit’ test(Bell et al., 2000). In this sense, the calibration is the process of getting the set of parameters through which the best-fit result could be obtained. Evaluation indicates the measurement of how good the fit is.

Owing to the complexity of the behaviour pattern of CA model, there are rare general automatic calibration techniques available now if not none. Manually calibration judged by visual interpretation has shown its feasibility in CA model calibration and remains as the major calibration method though it is very tedious. Evaluation is one constituent of the calibration process. The evaluation of the goodness of fit has to be carried out after the change has been made to the parameter settings and the evaluation result sequentially serves as the input information according to which the new parameters settings are set for the next model calibration.

In this section, the ZSSIM model has to be calibrated according to the empirical data obtained from field adopting the manually calibration approach. The result of the model will be evaluated by comparing them with the real development and this information in turn feed back to the calibration procedure. Finally, when the best-fit result has been generated from the model ZSSIM, it has to be evaluated in terms of accuracy of matching to the real city.

5.6.1. Calibration approach

A try and error approach was use to calibrate ZSSIM model. Visual interpretation is used to make the judgement on how good the model result is and again by visual interpretation, those factors who are thought to be too weak will get more weight and those too strong get less in the next run of simulation.

When calibrating the ZSSIM model, three general principles are followed:

- Start the model with the simplest case
- Change the balance of the factors
- Change the definition of model component only if it is really necessary

The first principle indicates that the calibration is started at the extreme weight settings. The second one means the adjustment on weight systems and sub weight systems. The third one suggests that only change the model definition as less as possible unless for sure the result is affected by the definition of model components.

When evaluating the results, the attention is focussing on the areas generated where the CA model is clearly wrong. Then by means of visual interpretation, the possible weak of strong factors will be found and adjustment on weight system will carried out accordingly. Finally, when the acceptable result has been generated, a precise evaluation based on cell-to-cell match is applied to give an overall accuracy in terms of exactly match of model result and real city. That is, to measure how much area ZSSIM model predicts as urban and which in reality also became urban, how much area ZSSIM model predicts urban, but in reality did not become urban, and how much area ZSSIM model did not predict urban, but in reality did become urban.

5.6.2. Scenarios generated by extreme parameter settings

Starting from the most extreme weight setting that almost all weight is set to one of the factors, some scenarios were generated. Based on the extreme scenarios and their comparison to the real developments, we can tentatively draw some conclusion that some of the factors are more affecting than the others. Then we can set more weight on these factors and use trial and error method to find the best-fit result.

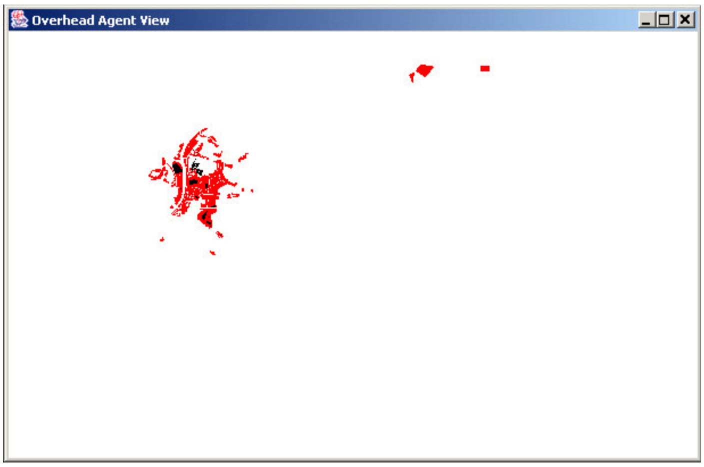


Figure 18 The urban form in year 1980

Figure 18 shows the urban form in 1980, which is the start point of model simulation. The red colour denotes the urban areas. The main urban centre is in the west part of the study area and the new centre, though still a village at that time, in the east part of the city.

Given the weight systems and all the factors and rules have already been incorporated into the CA model, first the most extreme parameter setting have to be tested to generate extreme scenarios. Then these scenarios have to be examined in terms of factor influence effect degree.

Table 8 weights setting for extreme scenarios

Weights			
Neighbourhood	Urban form	Road network	Terrain
0.9	0.03	0.03	0.04
0.03	0.9	0.03	0.04
0.03	0.03	0.9	0.04
0.03	0.03	0.04	0.9

Table 8 shows the weights setting for each affecting factors in the following simulations. The neighbourhood size used in these scenarios is 5 by 5 Moore neighbourhood. Land consumption is 459 cell per year in the first period and 946 cells per year in the second simulation period.

- Scenario 1: Neighbourhood effect 0.9(Figure 19)

Model year 1992 Model year 1998

Figure 19 High weight on neighbourhood effect

Under this parameter setting, the city will grow only at the adjacent area to the already developed area. (see Figure 19)

- Scenario 2: Urban form 0.9 (Figure 20)

Model year 1992 Model year 1998

Figure 20 high weights on urban form effect

Under this setting, urban form effect is the dominant influence and the urban will only growth in the locations where it is near to the city centre and at the same time it is allowed to be developed and it is previous vacant land. The sub weight system of urban centres ate set to equal weight.(see Figure 20)

- Scenario 3: Road network 0.9(Figure 21)

Model year 1992 Model year 1998

Figure 21 high weights on road network weight scenario

Under this setting, road network effect is the dominant influence and the urban will grow first along the roads. The sub weight system of road network was 0.7 to main roads and 0.3 to new developing zone centre.(see Figure 21

- Scenario 4: Terrain constraint 0.9 (Figure 22)

Model year 1992 Model year 1998

Figure 22 high weights on terrain constraint

Under this setting, terrain effect is the dominant influence and the urban will grow first in the flat areas. This one is quite similar to the real development that indicates that the real urban development is possibly influenced heavily by terrain constraint.(See Figure 22)

The scenarios generated from extreme weight settings imply that the real urban developments are influenced by all of these four factors. Among them, the road network and terrain seems have more influence then the other two factors. This will be the starting point of calibration for the best fit. The weights for road network and terrain constraint will be set relative higher then the rest factors when the calibration starts. Then according to the results generated from model, all the weights be adjusted and the model runs again using new weight system.

5.6.3. Sensitivity analysis

"Sensitivity analysis (SA) is the study of how the variation in the output of a model (numerical or otherwise) can be apportioned, qualitatively or quantitatively, to different sources of variation at the input, and of how the given model depends upon the information fed into it"(Saltelli et al., 2000). This defines a purpose in terms of what is done and "SA is a prerequisite for model building in any setting, be it diagnostic or prognostic, and in any field where models are used"(Saltelli et al., 2000).

By conducting sensitivity analysis, the following aspects should be investigated:

- How the output of a model varies as a function of variation in the input data and the model parameters.
- How does this model work - does it produce the effects that we expected?

In this sense, the sensitivity analysis could examine the significance of the model parameters. Sensitivity analysis varies parameters to see if they have big impact on output. If they do not, they may be considered to be removed. In this research, the sensitivity analysis was carried out only on the neighbourhood size. The above four scenarios generated by different weight settings could be considered as part of the sensitivity analysis.

Based on the same weights settings, the model was tested under different neighbourhood size settings.

Basic settings:

The weight system of MCE transition rule is set as:

```
      weight_Neighbourhood = 20,
      weight_urbanForm = 20,
      weight_Roadnet = 30,
      weight_terrain = 20,
      weigh_Masterplan = 10

The sub weight systems for road network and urban form are set as:
      weightMainroad = 0.7
      weightSecondaryroad = 0.3

      weightMainCenter = 0.0 from 1980 to 1992
      weightSubCenter = 1.0  from 1980 to 1992
      weightMainCenter = 0.5 from 1992 to 1998
      weightSubCenter = 0.5 from 1992 to 1998
```

Under this setting, the model was tested using four neighborhood sizes, namely small, middle, large, and very large which respectively indicate 3 by 3, 5 by 5 and 11 by 11 and 21 by 21 cells neighborhood. From small to very large, the cell considered its very adjacent neighbors to quite far neighbors.

- Neighbourhood definitions

Moore neighbourhood definition has been used. Following is the Moore neighbors (while squares) with the distance of one cell from the origin (shadow grid).

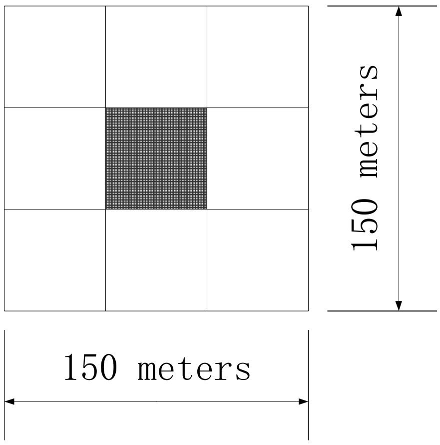
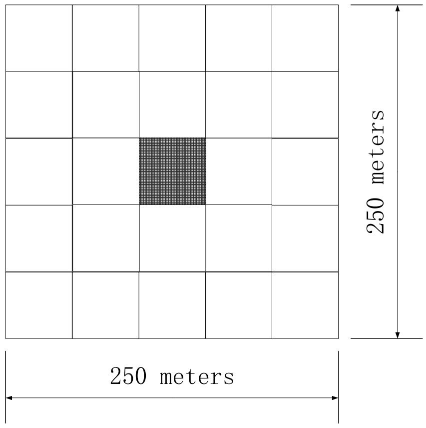
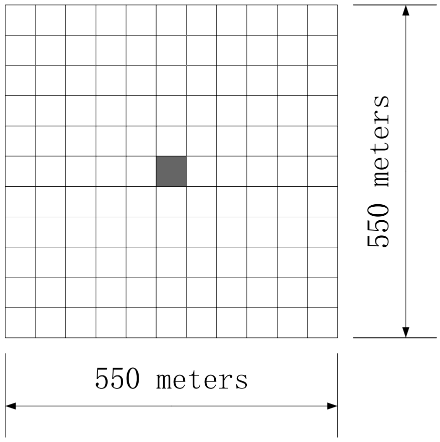
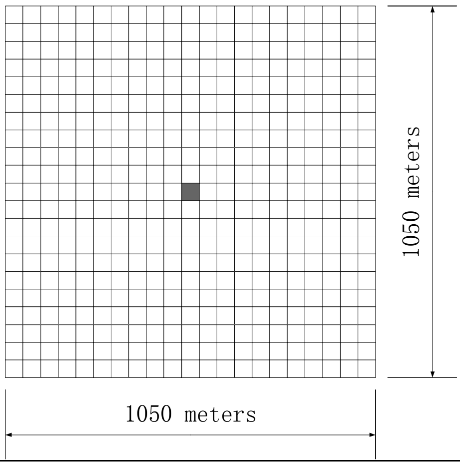
	
A: 3X3 neighbours	B:5 X 5 Neighbours
	
C: 11X11 neighbours	D:21 X 21 Neighbours

Figure 23 More Neighbourhood definition examples

- Simulation results of the different neighbourhood size in two different simulation periods:

	
3X3 in 1992	5 X 5 in 1992

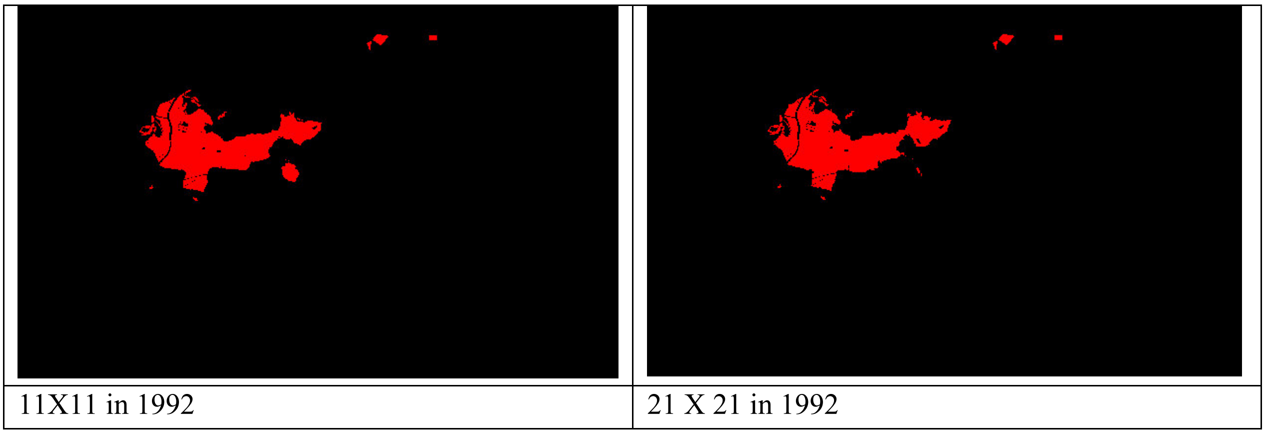


Figure 24 Sensitivity analysis on neighbourhood size in first simulation period

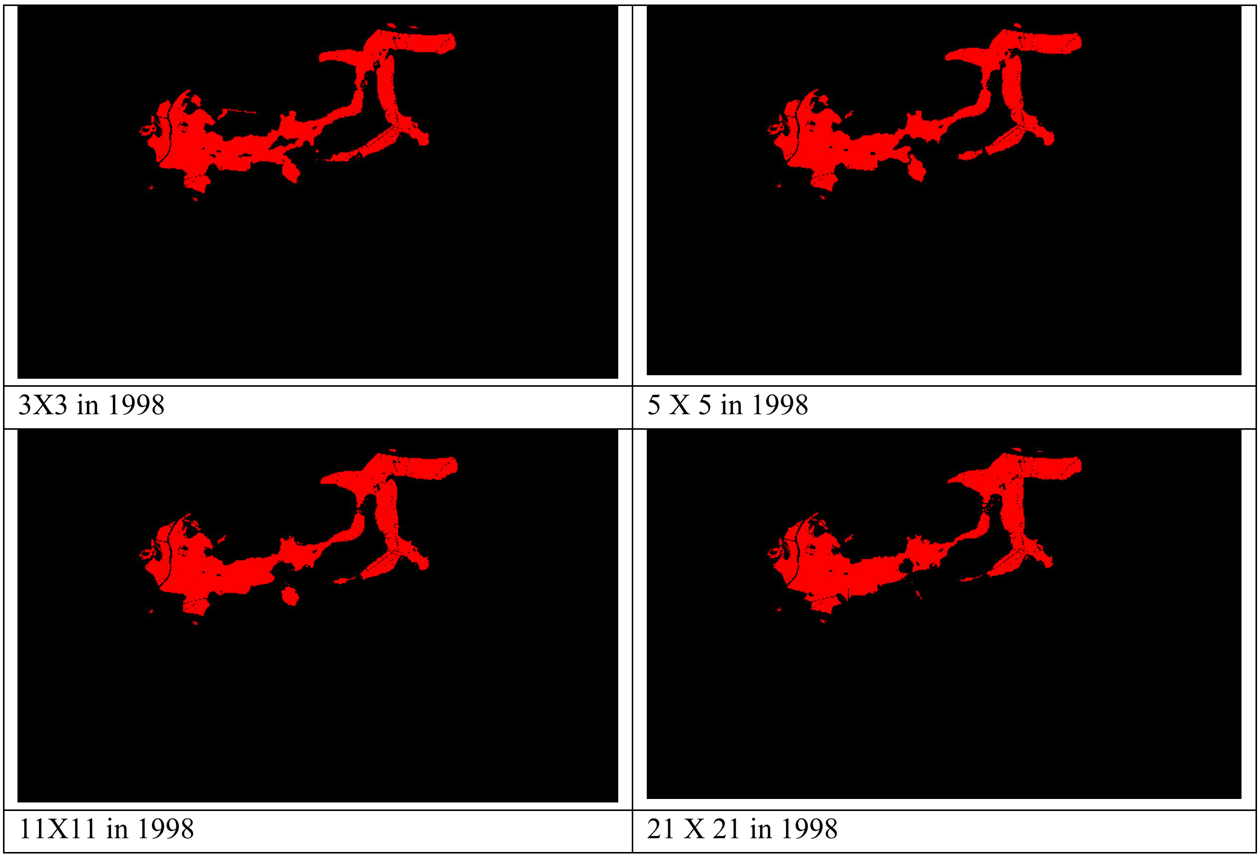


Figure 25 Sensitivity analysis on neighbourhood size in second simulation period

The results show that the neighbourhood size is not very significant in its variation by means of visual interpretation. However, it does show that the large size neighbourhood definitions lead to more agglomeration development forms comparing to the small neighbourhood size. Large neighbourhood tends to result in more compact urban development forms. When the cell considers its far neighbours, the urban grows more compact.

Since the sensitivity analysis does not shows significant differences in the simulation result from different size of neighbourhoods, the size of neighbourhood will be set to five (Figure 23 B) in the following calibrating procedure.

5.6.4. Best fit attempt

After the extreme scenarios are examined and the sensitivity analysis is tested on the effect of neighbourhood size, the next step is straightforward to take the attempt to simulate the real development. Following the try and error approach, the weights settings for affecting factors are adjusted for the next model run according to the previous model result.

The calibration was carried out in the following way:
From the results of those extreme scenarios, the road network attraction and terrain limitation seems more significant than the rest factors. By giving them relative high weights, say road network 0.4 and terrain limitation 0.3, and others 0.1, 0.1 and 0.1, the model runs once and results a scenario. This simulation result is then compared with the real situation and evaluated by visual judgement. From the visual interpretation, we can get some clue of which factor has too much influence and which has less effect than it should, then the less effect one's weight will be increased a little, say 0.05 and the too much effective one decrease its weight by the same value, 0.05. Gradually, the weight system changes little by little and the result moves towards the best fit.

The final weight setting used by the best fit scenario is:

- Road Net: 0.3
- Neighbourhood: 0.1
- Urban form: 0.1
- Terrain: 0.3
- Master plan: 0.2

Model year 1992	Model year 1998

Figure 26 Development generated by model

Figure 26 is the best fit result generated by the weights setting above. To examine how good the fit is, next section, the evaluation of this result, will give the answer.

5.6.5. Accuracy evaluation

This section analyse the accuracy of the best-fit model result. Figure 27 shows the real urban form in 1992 and 1998.

	
Real development in year 1992	Real development in year 1998

Figure 27 The real development of Zhongshan

By visual comparison, the model results are still not very similar to the real urban form. The real city growth is a dispersal form while the model gives a relatively compact developing style. Then we measured how much the difference between model result and real development.

Figure 28 Difference between model development and real development in year 1992

In Figure 28, the black colour denotes the unmatched urban cells in the model world and the real city in year 1992. The map is calculated by an exclusive OR operation between the model-generated map and the urban form map of real development in 1992. any urban cell in either model map or real map, if it can find a matched urban cell in the other map, it is set to white colour, if it can not find a correspondence urban cell, then it is set to black.

Figure 29Differences development from model and real city in 1998

Figure 29 The black area denotes the differences between the model development and real development in year 1998. The same exclusive OR operation is used here.

Since the land consumption was used to control the development speed in ZSSIM model, it makes no sense to compare the total developed cells by model with the real development. The comparison was calculated by comparing the consistent cells both in model development and in real development. Then we calculate four kinds of numbers:

- Matched urban cells
- Matched non-urban cells
- Urban cell in mode development but non-urban cell in reality
- Non-urban cell in model development but urban cell in reality

Based on the table cross in ILWIS, two error matrixes had been established as follow for two simulation periods:till 1992

The lattice has 606 * 376 cells, which equals 227856 cells. This number will be used as a base of the following accuracy calculations.

Table 9 Error matrix for first model period from 1980 to 1992

Real / Model	Urban	Non-urban	Total	Error of commission	User accuracy
Urban	3928	3170	7098	44.7	55.3
Non-urban	3157	217601	220758	1.4	98.6
Total	7085	220771	227856		
Error of omission	44.6	1.4			
Product accuracy	55.4	98.6			

The total urban cells in 1992 are 7098 cells in study city and the ZSSIM model developed 7085 urban cells till 1992. Non-urban cells in real city are 220758 and 220771 in ZSSIM model. Among the 7058 urban cells developed by ZSSIM model, 3928 of them matched with the reality. There are 3170 non-urban cells in the reality have been developed into urban cells. Therefore, the User accuracy is 3170 / 7089 = 55.3% and the Commission error of urban cells is 3928 / 7089 = 44.7%. Among 7085 urban cells developed by ZSSIM, 3928 of them matched with reality, 3157 of them did not become urban cells in reality. Thus, the product accuracy is 3928 / 7085 = 55.4% and error of omission is 3157 / 7085 = 44.6%. The same calculation applied to non-urban cells and resulted in Table 9.

The over all accuracy is 97.2%. It is calculated as: (3928 +217601) / 227856 = 97.2%, the sum of the matched urban cells and non-urban cells divided by the total cells number in the lattice. Since the non-urban area did not take any transition, this calculation can only give a rough idea about how good the match is. For this research, the ZSSIM model only develops non-urban cells into urban cells and the concern of this study is on the urban form, so the use accuracy of urban cells is the appreciate measure of the goodness of fit. It is 55.3% for the first model period.

There are 1960 urban cells in 1980, which should be excluded in the calculation because now we only concern the new developed urban cells real world as well as in model world from 1980. The comparison of the matched new developed urban cells in real city to the model world can give the idea about how the goodness of fit is. After exclude the original urban area 1960 cells from year 1980, the new development cells' user accuracy is (3928-1960) / (7098 – 1960) = 38.7%.

The same calculation applied in the second simulation result from 1992 to 1998, the following table shows the results.

Table 10 Error matrix for first model period from 1992 to 1998

Real / Model	Urban	Non-urban	Total	Error of commission	User accuracy
Urban	5166	6850	12016	57.0	43.0
Non-urban	5634	210206	215840	2.6	97.4
Total	10800	217056	227856		
Error of omission	52.2	3.2			
Product accuracy	47.8	96.8			

In Table 10, the total urban cells in reality are 12016 cells while 10800 cells in model result. Among the 10800 urban cells developed by ZSSIM model, 5166 of them matched the reality. After applied the same calculation in Table 9, the user accuracy and product accuracy for urban cells are 43% and 47.8% respectively. Over all accuracy is (5166 + 210206) / 227856 = 94.5%. After excluding the original 1960 urban cells in 1980, the user accuracy for urban cells in second simulation period is (5166-1960) / (12016- 1960) = 31.9%.

The over all accuracy is not surprisingly quite low. The reason partly subjects to the fact that there is no enough time to fine tune ZSSIM model to generate better results. This also implies that either more systematically and more adequately calibration should be carried out or more suitable model structure should be adopted or more rich data to feed into CA model in terms of social economical perspective. Discussion on this topic is presented in Chapter 6.

5.7. Visualization of the urban growth process

The urban growth is a dynamic process rather then a static structure. Owing to the flexibility of Java language and Ascape platform, it is quite easy to visualize the model result in Ascape or in animation files.

In the research, the visualization part is done through a file serialization process. In each model step, the current city form is recorded as a GIF file. These GIF files are processed in Ulead GIF animator software. Implementing in this way, the animation can be fully controlled in terms of speed and other animation effects.

Five animations had been made in this research: Neighbourhood effect animation road network effect animation, urban form effect animation, terrain constraint effect animation and best fit animation. These files can be found in the attached CD.

6. Discussion and conclusion

This study has explored the procedure of setting up a CA model to simulate the urban development process. Before we go to conclusion, it is worth to discuss about the CA modelling issues come across in this study and finally, recommendations for further work will be presented.

6.1. Discussion

In this research, effort has been put on constructing a CA urban model to simulate the urban growth process with trend break. Research questions have been set in chapter 1 to achieve the research objectives. These questions can be classified into four categories:

- What kind of CA model structure is suitable for this task?
- What kind o data is suitable for simulate urban growth in the study city?
- How to calibrate and evaluate the model?
- How to represent trend break in the model?

During this research, some of the findings and problems encountered in this research are discussed. In following sections, four aspects corresponding to the questions are discussed based on the results of this study. They are: CA model structure, data and scale, calibration and evaluation, external events. Another consideration that may have effect on the model result is dynamic interaction, it are also discussed at the end.

6.1.1. Model structure

A MCE controlled CA model was built under Ascape developing environment in this research. In ZSSIM model, cells are defined with two mutual exclusive states, urban and non-urban, and transition rules as a MCE process and are applied homogeneously to the whole city.

Whilst is not realistic to look the urban development as a homogenous process. In real urban context, the developments are happening in a heterogeneous manner. Usually, the new urban centre and new developing zones will have faster growth rates than the old centres. In addition, different kind of development has their own specific land requirement and different consideration to those influencing factors. For instance, industrial land use conversion concern more on transportation network more than neighbourhood effects and terrain limitations while residential land development concern more about neighbourhood effects and the trade off between land price and accessibility. In this sense, it is straightforward to use a fine defined land use category rather than a simplified urban non-urban state definition. Will more classes of land use type help? Considering the different land use type which naturally lead to different land requirement and different developing pattern, we could assume that using different transition rules to each of these different land use type is much reasonable than using only urban and non-urban land use category.

Another concern is that the urban areas are not built up immediately. From the constructing to built up, it takes some time to finish the conversion. This research also implemented constrained "grey cell" proposed and tested by Yeh (Yeh and Li, 2001). (See Appendix 2) In this approach, the cell experiences a mature process, in which it cumulates the developing potential value till it reach a certain threshold, to transit from non-urban to urban. Due to the difficulty in interpretation of the results and synchronization of time, it was abandoned finally. However, this implementation shows some potential to be further improved since the mature process represents the real development of the city naturally.

In ZSSIM, the transition rules are designed as a multi criteria evaluation process. Thus, the manipulation on the weight system could generate interpretable result. One of the criteria in this MCE is worth to discuss here, neighbourhood effect. Effect of neighbourhood is always positive in terms of contribution to the overall assessment. Whilst it is possible, that too much surrounding developed cell lead to a negative impact on the developing potential vales. In the recommendation section, the suggestion of defining the negative neighbourhood effect is given.

6.1.2. Data and scale

In CA urban model, data is always parsimony. It is thought that the more data input, the higher possibility of getting a better result comparing with the studying world though at some point, an extra data has very less added value to the simulation.

Data adequacy

Technology has improved a lot in last two decades and the information management has achieved a significant achievement in most developed western countries. However, it is not the same case in the developing countries. Thus choosing affecting factors becomes extraordinarily important in modelling urban growth process. Like many other simulations, ZSSIM model is an abstraction, which uses only a small number of the factors involved in the real physical world. However, the phenomenon of urban sprawl is far more complex and this has only been an attempt to investigate a few aspects of the system dynamics within a city.

The historical data management is poor in China. It is difficult to get the land use map and social economic information even for current situation let along in time series. It will be helpful to incorporate social economic data into the CA urban model. These social economic aspects influence urban development heavily. For instance, population density is a crucial factor, which influences the site selection and developing speed for a particular area. Other social economic data like income level will at some extent determine the mobility of the residents which sequentially influence the consideration of neighbourhood effects in terms of difference distance decay curves.

This study only uses the transportation, urban form, terrain limitation and master plan factors. Without social economic factors inside, the model is neither complete nor realistic though many researchers argue that CA model could use some key factors generating quite realistic phenomena. Whilst as discussed above, limited by the data availability and the nature of enormous data requirement in CA modelling practice, we can only use as much the data as we can get and try to get the best result from it.

From the result scenarios generated, we can see that the match to the reality is relatively low. One possibility is that dealing this urban growth issue in such a relative large scale isolated the study area with other small towns surrounding this city. In reality, the urban planners, also the land investors will concern the administrative boundary not violating or not exceeding the municipality administrative boundary. Different towns around the study area have its distinct influence on the target city.

In the context of Zhongshan city, its affiliate towns have different social economic characteristics. Some of them are mainly the agriculture function and some of them are light industrial towns. The city tends to avoid the merge with its very near neighbours. To take this prospect into account, the CA urban model should consider the surrounding towns and this sequentially lead to use a relatively small-scale map representation.

Spatial resolution

Another issue is that when we make a model at large scale, the unit of the model represents a small area or disaggregated individuals, which is naturally difficult to simulate if it is not impossible. It is thought that modelling population or household behaviour is more suitable to use a multi agent approach rather then CA approach.(Parker et al., 2001) Modelling individual behaviour involves mimicking the intelligence behaviour of human beings, which is not a feature of CA model but a feature of multi agent system. A cognitive agent can react to the stimuli from its environment and try to achieve its own goal, find a settlement or find a path, etc.

Using CA model modelling the urban development needs a carefully consideration of choosing exact scale, namely spatial resolution. Meaningful resolution makes sense of the simulation results. Since the CA urban model is modelling the individual spatial unit evolves over time, the unit has to be meaningful to consider the reasonable transition rules for it. In this research, the spatial resolution is set to 50 meters, which is almost the normal size of a building in Zhongshan city. Choosing a large size as a block size also makes sense.

Time resolution

Time resolution is also crucial in this kind of research. How long should the time intervals be? In ZSSIM model, the time interval is set to one year. The same consideration should be carried out as the consideration in spatial resolution, make it meaningful. The construction duration of a new residential building usually takes about one year in China. Industrial and commercial areas take longer time then housing area to finish the construction. Thus, ZSSIM model use one year as time resolution to reflex the construction duration of residential areas.

From another point of view, we could also conceive the urbanization process is just start from the starting of a site construction. Hence, it is possible to use a fine-grained time scale such as one month to represent the development.

6.1.3. Calibration and evaluation

Calibrating a CA urban model using empirical data is still an issue on debate. How good is good enough in terms of an operational CA model? Which approach should the model builder use or is there standard method to calibrate this kind of model?

From the analysis in chapter 5, the scenarios generated by this CA model can only get quite limited similarity to the reality, which is quite normal in this kind of modelling effort. Following are some aspects that may contribute to this low matching. In the following section I will present several considerations arising from the presented work.

For measure the similarity of the model result and real city, a coincidence matrix which is generated by cell to cell comparison of two maps is used in this research. . However, urban systems are rather complicated and their exact evolution system is unpredictable. It is argued that CA simulations should not be assessed just on the goodness of fit but also on feasibility and plausibility, that is, on possible urban forms.(Yeh and Li, 2001)

The accuracy in chapter 5 is quite low which also subject to the method used to access the accuracy. When studying the urban form, pattern is more important then the how many matched cells of model and reality. In this sense, the accuracy should not only take those exactly matched cells but also the spatial similar cells into account. Possible alternatives include fractal dimension methods. Fractal dimension is a measure of how "complicated" a self-similar figure is. It can be used to reflect how much space is filled. A larger value of the dimension means that the city is more compact. Yeh (Yeh and Li, 2001)has used a fractal base method to evaluate the model performance. He used a density-radius relationship based method to measure the fractal dimension.

6.1.4. External events

CA model is supposed to be a self-organized system, which means there should be no interference from outside. Whilst modelling the urban growth, which is heavily influenced by the decisions made by people, the outside interferences are always exist and play an important role. Policies changes, incidentally influences, social economic situation dramatic changes, etc, all of these will cause the model loss its ability to mimic the reality. These external events, if their consequences are significant, will lead to trend break in the urban development.

As discussed in chapter two, the modifications have to been done when modelling the real world. Nevertheless, how to incorporate these issues into CA model are still under debating. Batty (Batty et al., 2002) has built a multi agent model to simulate a parades movement in London, which had tackled the external interferences. In ZSSIM model, the trend break is represented by using a new set of control parameter setting and the new road network data input in the middle of the simulation. Configuration in this way, ZSSIM model in some extent introduces the external events into CA model.

Despite the fact that, Batty (Batty et al., 2002) still stated that though the external events could be taken care by the intelligent agents in some extent, it is yet better to be controlled by the modeller himself.

6.1.5. Dynamic interaction

In this research, to reflex the trend break in the middle of the simulation, the ZSSIM model managed to used different control parameters and new input data through a central control mechanism, root agent. This has introduced some dynamics interaction features into this model.

Whilst in reality, most of the factors identified in chapter four are interacting with each other in different ways. The urban form itself definitely affects on road network. A high population density with insufficient accessible road net will strongly demand for the construction of new road to improve the accessibility for that area. New urban development tends to happen nearby the new road network where it has good accessibility as well as a relatively low land price. After massive of residential or industrial land construction, the road network in this area will be no long sufficient. This leads to a new demand of road construction. Efforts on modelling urban in a truly dynamic way are very indispensable in consideration of modelling the reality as real as possible.

To model the urban growth in a truly dynamic way, using only CA itself seems not sufficient to achieve this goal. Using the same example above, the roads construction is obviously not a self-organized process. Secondary roads always show their effects later then main roads because usually they will be construct later after the main roads finished. In this research, the developing of road network was represented by the changing of road network layer by the root agent in the middle of the model running. But this only roughly represented the road network change process, fine scale new road effect approach still need to be investigated.

6.2. Recommendations on future reseach

Based on the exploration on CA urban growth model and the discussion above, recommendations on the further research are given in this section. These present the aspects of the model that should be developed further and directions in which further research would be desirable.

6.2.1. Model improvement

There are several aspects of ZSSIM that should be extended and adapted in order to improve it. These are negative neighbourhood, variable transition rules, social economic data input. These issues are addressed in more detail below:

- Negative neighbourhood function

It will be interesting to use a negative neighbourhood function as well. In urban growth process, there is not only a consideration of developing nearby the built up area, but also the consideration of avoiding over crowd. Hence, it is natural to also define the negative effect of neighbourhood. For instance, the neighbourhood effect can be define as this, for one cell, if the surrounding cells are less then a certain number, then the influence is positive, if the surrounding cells' number exceed a certain number, it will be too crowed and the effect of neighbourhood will be set as negative.

- Variable transition rules

From the discussion in previous section, urban development is not happening homogeneously. Thus it is reasonable to use different transition rules for different land use types and different transition rules for different areas.

When use land use type specific transition rules, we have to consider the different preference of land characteristics for different land use type. For example, residential lands need good accessibility and low price, industrial land use need good accessibility and good utility services, commercial land use concern the population density and income group rather then land price, all these considerations lead to define different evaluation approaches for different land use in transition rules. Moreover, different land use may use different neighbourhood size since they have different consideration of neighbourhood effect.

Another aspect is that different areas in a city may have different developing speeds. Thus, we could use different time intervals for different areas to control the transition speed. For example, in area A, each model step equals to one year in reality, while in area B, we can set each model step equal to 3 months, thus the different developing speed can be incorporated into one CA model.

6.2.2. Further research

From the discussion above, especially the literature review, several aspects of the model were able to improve without rather radical change. Further explorations on different model structure and model control mechanism would contribute to enrichment of urban model methodologies.

- Hybrid model structure

As Torrens argued, single CA model may not be sufficient to model the whole urban dynamic, to make the model more flexible, MAS and traditional model could be integrated into one model within which the output of one model is the input of other model, models interacting with each other in a co-operative way. It will be interesting to use a Hybrid model in such a way that the mobile agents representing the different stakeholders in urban development playing around in a CA environment which itself evolves in time. To do this, the social economic information about the study city is crucial.

Human like agents can be adopted in the model that each type of them represents a actor in the real world. For example, we can use a type of agent to represent housing construction contractor, entrepreneur agent to represent entrepreneur, habitant agent to represent residents. They will move around the space and pursue their own goals. Construction contractors will search for the low price land which is near to city centre or near to roads and construct new house, habitant agents will search for new house which is near to working opportunities, entrepreneur will search for industrial sites according to land prices and other criteria. When these agents act on their environment, the environment itself also evolves over time, it can be configured as a CA space.

Developing urban model like this will be extremely interesting and may generate interesting result.

- Integration with fuzzy-set approaches

When we define the neighbourhood effects, we use terms as small large, etc. When we talking about neighbourhood effects, we measure the influence in terms of adjacent, nearby cells, etc, which are naturally fuzzy concepts. How far is far and how far is near? When we define the developing speed of different zones in the city, the same thing happens to the time aspects, we also use terms as slow, fast, etc. People describe urban development using words such as “rapid developing”, “slow growth”, etc. Other aspects such as the definition of urban boundary between an urban built up area and its non-urban hinterland, etc. These definitions are quite fuzzy. It was thought that using a fuzzy-set approach for modelling urban development is appreciated and will generate more realistic results. (Yan and Phinn, 2002)

6.3. conclusion

The main objective for this research is to develop a CA model that could represent the different trends of temporal-spatial pattern for a case study in China for different time periods.

7. To design a CA model structure to simulate the urban form change process for the study city
8. To define model elements
9. To prepare the necessary data to be used in CA simulation model
10. To implement the CA model prototype and calibrate the prototype
11. To present the different developing trends in two different time periods
12. To validate the established CA model

In this research, a CA model, ZSSIM, for simulating the urban growth process in a case study city, Zhongshan, China, has been developed and implemented within Ascape multi-agent developing environment using JAVA language. This model uses a multi criteria evaluation (MCE) transition rule and a root agent, which is a kind of central control mechanim, together to control the non-urban to urban transition process.

Five factors were considered to have impacts on urban growth: neighbourhood effect, road network, urban centres, terrain constraint, and policy. The effect of these factors was evaluated by the MCE transition rule and result in a developing potential value, which indicates the transition potential. The root agent ranks all the potential values into descend order and get a threshold for cell transition according to the annually land consumption. Finally the cells are transited according to their ranking positions compared with the threshold.

To simulate the trend break, a central control mechanism was introduced into this CA model. The root agent was defined to take care of the model data and parameters.

The model was calibrated through a manually calibration process and accuracy assessment of the best fit result was evaluated in a cell-to-cell match approach. Finally, the simulation result was exported into animation form which visualized the growth process.

Though the accuracy of best-fit is not yet good, CA model using MCE transition rules and with central control mechanism, can in some extent simulate the urban growth process with trend break in the middle. Thus, conclusions can be addressed as:

- CA model using MCE controlled transition rule is a useful tool for modeling urban growth
- ZSSIM can be a basis for further development of a tool for studying and predicting urban growth

process

- Modeling trend break, which is usually caused by top down process, needs to introduce central control mechanism into CA model
- Manually calibration and visual evaluation are acceptable and feasible approaches in CA urban grow models

In summary, most of the objectives have been achieved through this research.

7. References

Batty, M., DeSyllas, J. and Duxbury, E., 2002. The Discrete Dynamics of Small-scale spatial Events: Agent-based Models of Mobility in Carnivals and street Parades, Centre for Advanced Spatial Analysis, London.

Yan, L. and Phinn, S.R., 2002. Modelling urban development with cellular automata incorporating fuzzy-set approaches. Computers, Environment and Urban Systems(358): 22.

Yeh, A.G.-O. and Li, X., 2001. A constrained CA model for the simulation and planning of sustainable urban forms by using GIS. Environment and Planning B: Planning and Design, 28: 733-753.

Barros, J. and Sobreira, F., 2002a. CITY OF SLUMS: SELF-ORGANISATION ACROSS SCALES, International Conference on Complex Systems (ICCS2002), Hashua,NH,USA.

Barros, J. and Sobreira, F., 2002b. City of Slums: Self-organization across scales. 55, Centre for Advanced Spatial Analysis University College London, London.

Batty, M., 1998. Urban evolution on the desktop: simulation with the use of extended cellular automata. Environment and Planning A, 30: 1943-1967.

Batty, M., 2000. GeoComputation using cellular automata. In: S.O.R.J. Abrahart (Editor), GeoComputation. Taylor &Francis, London, pp. 413.

Batty, M., DeSyllas, J. and Duxbury, E., 2002. The Discrete Dynamics of Small-scale spatial Events: Agent-based Models of Mobility in Carnivals and street Parades, Centre for Advanced Spatial Analysis, London.

Batty, M. and Torrens, P.M., 2001. Cellular automata and urban simulation: where do we go from here? Environment and Planning B: Planning and Design, 28: 163-168.

Batty, M., Xie, Y. and Sun, Z., 1999. Modeling urban dynamics through GIS-based cellular automata. Computers, Environment and Urban Systems, 23: 205-233.

Bell, M., Dean, C. and Blake, M., 2000. Froecasting the pattern of urban growth with PUP: a web-based model interfaced with GIS and 3D animation. Computers, Environment and Urban Systems(24): 559-581.

Besussi, E., Cecchini, A. and Rinaldi, E., 1998. The diffused city of the Italian North-east: identification of urban dynamics using cellular automata urban models. Compt, Environ. and Urban Systems, 22(5): 497-523.

Cechini, A. and Rinaldi, E., 1999. Building Urban Models with Multi Cellular Automata, the 6th international conference: computers in urban planning & urban management, Venice, Italy.

China, E.i.c.o., 2003. China environment protecting network.

Coucleis, H., 1997. From cellular automata to urban models: New principles for model development and implementation. Environment and Planning B: Planning and Design, 24: 165-174.

Department for Transport, L.G.a.t.r., 2003. Multi Criteria Analysis: A manual.

ESRI, 2003. ARC/INFO online help. ESRI.

Gardner, M., 1970. the fantastic combinations of John Conway's new solitaire game "life". Sceientific American(223): 120-123.

Gilbert, N. and Terna, P., 1999. How to build and use agnet-based models in social science.

Grueau, C., Rodrigues, A. and Muge, F., 1999. A Multi-agent Approach For The Management Of Quarrying Environmental Impacts.

Itami, R.M., 1994. Simulating spatial dynamics: cellular automata theory. Landscape and urban planning, 30: 27-47.

Klaus, H. and Laixiang, S., 2001. Land Use Change at the National and Regional Level in China:A Scenario Analysis Based on input-output Modeling, International Institute for Applied Systems Analysis, Laxenburg, Austria.

Martin, D. and Wu, F., 1999. Empirical CA simulation from a high resolution population surface, the 4th International Conference on GeoComputation, Mary Washington College Fredericksburg, Virginia,USA.

municipality, Z., 1993. Master plan 1992.

O'Sullivan, D., 2000. Graph-cellular automata: a generalised discrete urban and regional model. Environment and Planning B: Planning and Design, 28: 687-705.

O'Sullivan, D. and Torrens, P.M. (Editors), 2001. Cellular models of Urban Systems. Theoretical and Practical Issues on Cellular Automata. Springer-Verlag, Karlsruhe,Germany, 108-116 pp.

Parker, D.C., Manson, S.M., Janssen, M.A., Hoffman, M. and Deadman, P., 2001. Multi-Agent System for the Simulation of Land-use and Land-Cover Change: A Review.

Parker, M.T., 2000. Ascape: Abstracting Complexity, Swarmfest 2000 proceedings forthcoming. Brooking website.

Parker, M.T., 2001. What is Ascape and Why Should You Care? Journal of artificial societies and social simulation, 4(1).

Pavard, B. and Dugdale, J., 2002. An Introduction to Complexity in Social Science.

Rennard, J.-P., 2000. Introduction to Cellular Automata.

Roger, L.S., Schwarz, P.A. and Johnston, K.M., 2002. Some Guidelines for implementing spatially explicit, individual-based ecological models within location-based raster GIS.

Saltelli, A., Chan, K. and Scott, E.M., 2000. Sensitivity Analysis [Willey series in probability and statistics]. Willey, New York.

Shenghe, L. and Sylvia, P., 2002. Spatial Patterns and Dynamic Mechanisms of Urban Land Use Growth in China: Case study in Beijing and Shanghai. IR-02-005, International Institute for Applied Systems Analysis, Laxenburg, Austria.

Takeyama, M. and Coucleis, H., 1997. Map dynamics: integrating cellular automata and GIS through Geo-Algebra. International Journal of Geographical Information Science, 11(11): 73-91.

Torrens, P.M., 2000. How Cellular Models of Urban Systems Work (1. Theory). 28, Centre for Advanced Spatial Analysis University College London, London.

Torrens, P.M., 2001. Can Geocomputation save urban simulation? Throw some agents into the mixture, simmer, and wait... Annual assosiation of American Geographers Meeting, New York City,USA.

Torrens, P.M. and O'Sullivan, D., 2000. Cities, cells, and cellular automata: Developing a research agenda for urban geocomputation., Fifth Annual Conference on GeoComputation, Manchester.

Voogd, H., 1983. Multi criteria evaluation for urban and regional planning. Pion Limited, London, 367 pp.

Wolfram, S., 1983. Cellular automata. Los Alamos Science, 9: 2-21.

Wolfram, S., 1984. Cellular automata as models of complexity. Nature, 31(4): 419-424.

Wolfram, S., 1988. Complex System Theory, Emerging Syntheses in Science: Proceedings of the Founding Workshops of the Santa Fe Institute. Addision Wesley, pp. 183-189.

Wu, F., 1998. Simulating urban encroachment on rural land with fuzzy-logic-controlled cellular automata in a geographical information system. Journal of environmental Management, 4(53): 293-308.

Wu, F. and Webster, C.J., 1998. Simulation of land development through the integration of cellular automata and multicriteria evaluation. Environment and Planning B: Planning and Design, 25: 103-126.

Yan, L. and Phinn, S.R., 2002. Modelling urban development with cellular automata incorporating fuzzy-set approaches. Computers, Environment and Urban Systems(358): 22.

Yearbook, C.S., 1999. China Statistical Yearbook, statistical office, Beijing.

Yeh, A.G.-O. and Li, X., 2001. A constrained CA model for the simulation and planning of sustainable urban forms by using GIS. Environment and Planning B: Planning and Design, 28: 733-753.

Zeigler, B., 1984. Theory of modeling and simulation. Robert E. Krieger Publishing Company,Inc., Malabar,FI.

Appendices

Appendix 1 Data Presentation

Following are the data gotten from field:

- Land use map: (from urban planning bureau)

1980 TIF file
1992 TIF file
1998 Microstation DGN files

- Master plan map: (from urban planning bureau)

1980 TIF file,
1985 TIF file,
1992 TIF file,
1998 Microstation DGN files

- Topographic map: (from land administration bureau)

1958 TIF file
1983 TIF file
1998 AutoCAD DWG files (including transportation network map)

- Population data: (from public security bureau)

From 1990 to 2002 annually (copies in tables) at administrative district level

- Land price data: (from land administration bureau)

2001(rough) on paper (tables)
Year 2002 in Mapinfo format

Auxiliary data:
Master planning documents (1980, 1985,1992,1998)
Yearbook in 2002

Appendix 2 "Gray cell" approach

This is complementary CA model I implemented at the beginning. This CA model adopted the model structure of Yeh's (Yeh and Li, 2001) model, that the cell does not change its state immediately after the transition rules has been applied. The approach was named as "Grey cell" approach which indicates that the cell will be develop into urban land use through a "white", "grey" and "dark" stages, which in turn represent the non-urban, partial urban, and urban states. The cell have a continuous "Grey cell" value from 0 to 1, which indicate the cumulative process. When the transition rules are applied on that cell, it just cumulates the development potential, after the cumulative potential value reaches 1, the cell change its state to 1 and stay at 1, developed. The potential development value of that cell is determined by the following formulae:

$$G_{xy}^{t+1} = G_{xy}^{t} + \Delta G_{xy}^{t} \quad (1)$$

Where G_{xy}^{t+1} denotes the potential of development in position x, y in time step t+1, G_{xy}^{t} denotes the value in time step t, ΔG_{xy}^{t} denotes the gain value from time step t to t+1 for the same cell.

The definition and representation of influencing factors are the same as the approach use in chapter four Except :

- Neighbourhood factors
-

Considering only the neighbourhood function, the potential value increment can be defined as:

$\Delta G_{xy}^{t} = f_{xy}(q, N)$

Where $f_{xy}(q, N)$ is a function of the neighbourhood for cell in location (x,y), q is the developed cells in the neighbourhood and N is the number of the neighbours. The simplest way to define this is to use the q as the amount of developed cells in the neighbourhood and use N as the total area represented by the neighbourhood. In this sense, f is define as

$f_{xy} = q / (N * A)$

Where A is the area represented by one the cell, q is the number of developed cells in the neighbourhood and N is the total cell number of the neighbourhood. More complex definition can introduce the direction control into the neighbourhood function using weight system in neighbourhood.

- Regional factors
-

Considering the impact of city centres and sub-centres, the Natural constrains and transportation network, the equation comes like this:

$\Delta G_{xy}^{t} = f_{xy}(q, N) \times \prod \delta_{xy}(F, N, T)$

Where f_{xy} (q, N) is the same as the last equation, $\prod \delta_{xy}(F, N, T)$ denotes the factors affect on urban form development, *F* denotes the urban form effects and *N* natural limitations. *T* is the transportation factor.

The function of $\delta_{xy}(F)$ is determined by the relationship between urban growth and urban centres. The same distance decay function was used as in chapter four.

$$F_i = \sum_{j=1}^{n} W_j e^{-\beta d_{ij}} \qquad \text{(ESRI)}$$

Where:
F_i is accessibility to urban centres at cell i
W_j is attractiveness of location j and here I just use it as the weight of the centre j
d_{ij} is distance between cell i and centre j
β is exponent for distance decay
n is number of centres in the city

The function of $\delta_{xy}(N)$ is those constrains which are set by the natural environment. Water body and terrain limitation are taken into account. The urban development usually tries to avoid changing the water system. This research assumes that urban development will not change the water body. Thus the only constrain to the urban expansion is that urban is not allowed to develop on the water area. The slope of terrain also affects the urban shape formulation but only those steep areas are not suitable to develop. This factor is also set as 1 “allowed”, flat area, or 0 “not allowed”, hilly area, for development.

The function of $\delta_{xy}(T)$ is the transportation network influence. Usually it is calculated in terms of accessibility. Without detailed population data and other thematic data, the accessibility is defined as the distance decay function of the distances to the main roads.

$$T_i = \sum_{j=1}^{n} W_j d_{ij}^{-\beta}$$

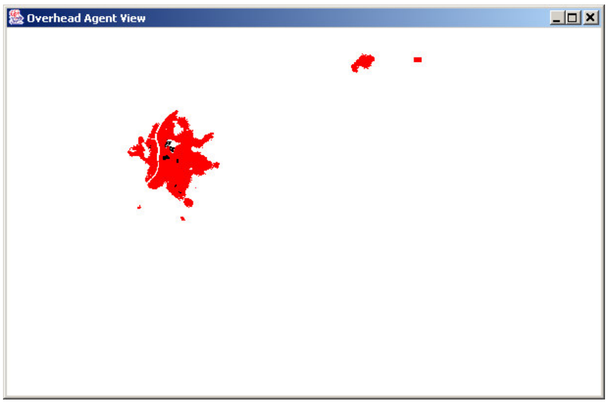
Where T_i is the accessibility of cell i, W_j is the weight for road class j, d_{ij} is the distance from cell i to the nearest road j.

- Stochastic factors
-

In urban developing process, always some unpredictable events impact the urban growth. To represent this disturbance, a stochastic parameter is introduced into this model:

$$\Delta G_{xy}^{t} = [1+(-\ln\gamma)^{\alpha}] \times f_{xy}(q, N) \times \prod \delta_{xy}(F, N, T)$$

Where $[1+(-\ln\gamma)^{\alpha}]$ is the stochastic disturbance, γ is a random variable within the range {0,1}, and α is a parameter to control the size of the stochastic perturbation. α can be used as a dispersion factor in the simulation.



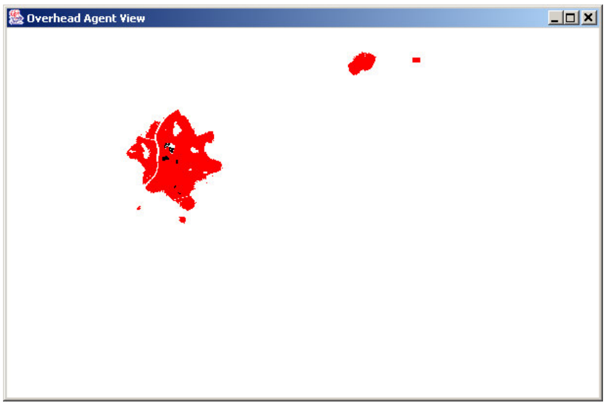


Figure 30 Urban forms for step 2159 and 3851 with stochastic degree 5

The simulation results show the ability of generating different urban forms in terms of compact, disperse under this models configuration. While some problem existed in this model is that using a non-compensative approach, it is difficult to manipulate the different effects of different factors as well as the difficulty in synchronize the model time to real world time. Owing to these reasons, the final model configuration was implemented in a compensative MCE manner.

Appendix 3: Performance consideration in Ascape

Ascape was designed as a flexible and robust general purpose multi agent developing platform. Whereas to implement a CA model in Ascape environment, some of the features specially design for multi agent system are no longer advantages. Considering the simplexity of CA model, comparing with multi agent models, in terms of model structure and individuals' behaviour, some modification and variation of Ascape were made during the development of ZSSIM urban simulation model.

By default, the agents in Ascape apply their rules randomly. And there are two execute ways: complete run and random run. Organized in this a way, Ascape prevents the sequential effects from sequential calculation on each agents. This is essential in a multi agent model in which the agents are mobile and interact intensively. Whilst in a CA model as ZSSIM, by carefully arrangement of the rules, we could safely run the model in a sequence way that the cells in the CA model apply the transition rule one after another. The typical case of setting this off is that the outcomes of the execution of this rule do not affect other agents and the agents ate not affected by execution order. Running the CA model in a sequential order could enhance the model performance significantly due to the removal of unnecessary random number generating process for each cell. To turn off this feature, I overrided the method isRandomExecution() in my rule definition part. Let isRandomExecution() return a false to tell the root model that this rule does not need to be executed randomly.

```
public final static Rule DEVELOP_POTENTIAL = new Rule("calculate the potentials") {
        public void execute(Agent agent) {
        //calculate potentials
                ((landCell) agent).setPotential( ((landCell) agent).calculatePotential());
                                                                                (((Lan-
                dUse)agent.getRoot())).addPotentials(((landCell)agent).getPotential());
        }
        public boolean isRandomExecution() {
                return false;
        }
        public boolean isCauseRemoval() {
                return false;
        }

};
```

The codes above also give an example of the way of defining rules. The method isCauseRemoval() set to return false also could improve the model performance because the conservative setting of Ascape assuming the behaviour of the agents will cause removal of other agents. In ZSSIM model, we set the rule be executed in sequential order and it do not cause deletion of other agents.

Another aspect of enhancing the model performance comes from the neighbourhood defines. To improve the model performance, Ascape stores the immediate adjacent neighbours for each cell in the lattice so that when counting the neighbours or searching for its neighbours, the cell can use its stored neighbours immediately. In ZSSIM model, the neighbourhood size were tested by different settings and the neighbourhood size may vary, thus there is no need to store the neighbours for each cell and by not doing this, the model can leave more memory for other cell attributes. This was done by modifying the Ascape source code, adding a control parameter "preSetNeighbors" to ScapeArray2d class, to control whether to set the neighbours(1 distance) or not. I set the preSetNeighbors to false so that it will not have inherited neighbours for the neighbourhood functions.

Appendix 4 Source code of ZSSIM in JAVA

Following are the classes defined in ZSSIM model. Those who have interesting in build their own CA models could modify it in whatever the way they want.

- Class ImageContainer

```
package zssim;

/**
 * @author szw
 *
 * To change this generated comment edit the template variable "typecomment":
 * Window>Preferences>Java>Templates.
 * To enable and disable the creation of type comments go to
 * Window>Preferences>Java>Code Generation.
 */

import java.awt.image.*;
import java.awt.image.renderable.ParameterBlock;
import java.io.*;

import javax.media.jai.*;

import com.sun.media.jai.codec.*;

import edu.brook.ascape.model.Cell;
import edu.brook.ascape.model.ScapeGraph;

import javax.media.jai.iterator.*;
import javax.media.jai.widget.ScrollingImagePanel;


     /**
      * This program decodes and encodes an image file of two JAI supported
      * formats, JPEG and File format. Used in LandUse Model.
      */
public class ImageContainer {
               int height;
               int width;
               double[][] max_min_Dn;
               String filename;
               RenderedOp loadImage;
               BufferedImage blankImage;
               String fileFormat = "JPEG"; //others like TIFF, BMP, etc. see JAI help


               boolean flag_Write = false;

               /**
                * Constructor for ImageContainer.
                */
               public ImageContainer(int w, int h) {  // in this case means to write
                       super();
                       flag_Write = true;
                       setWidth(w);
                       setHeight(h);
                       blankImage = new BufferedImage(width, height, BufferedI-
mage.TYPE_INT_RGB);

               }

               /**
                * Constructor for ImageContainer.
                */
               public ImageContainer(String string) {
                       super();
                       //pass the file name to file name
                       filename = string;

                       openfile();
                       height = loadImage.getHeight();
                       width = loadImage.getWidth();

               }

               public void openfile(){
                       //open the fileoperator to be used by import
                       /*
```

```
         * Create an input stream from the specified file name
         * to be used with the file decoding operator.
         */
        FileSeekableStream stream = null;
        try {
             stream = new FileSeekableStream(filename);
        } catch (IOException e) {
             e.printStackTrace();
             System.exit(0);
        }

        /* Create an operator to decode the image file. */
               loadImage = JAI.create("stream", stream);

               }

    /**
     * Import integer tiff map .
     */
    public int getMapValue(int x, int y) {

               /* Create the output image
               TiledImage outImage;

        outImage = new TiledImage(loadImage.getMinX(),
                                  loadImage.getMinY(),
                                  loadImage.getWidth(),
                                  loadImage.getHeight(),
                                  loadImage.getMinX(),
                                  loadImage.getMinY(),
                                  loadImage.getSampleModel(),
                                  loadImage.getColorModel());
               */

               // Loop over the input, copy each pixel to the output, modifying
        // them as we go

        //int bands = loadImage.getSampleModel().getNumBands();

        // used to access the source image and maybe use to import the image into latice
        RandomIter iter = RandomIterFactory.create(loadImage, null);


        //for (int band=0; band < bands; band++) {
            //for (int line=0; line < height; line++) {
                //for (int samp=0; samp < width; samp++) {

                    int dn = iter.getSample(x, y, 0);



             //   }
           // }
        //}

               return dn;

    }

     /**
     * Import map from float Tiff files
     */
    public float getfMapValue(int x, int y) {

        RandomIter iter = RandomIterFactory.create(loadImage, null);

        float dn = iter.getSampleFloat(x, y, 0);

               return dn;

    }



               /**
                * Returns the height.
                * @return int
                */
               public int getHeight() {
                       return height;
               }

               /**
                * Returns the width.
                * @return int
                */
```

```
		public int getWidth() {
			return width;
		}

		/**
		 * Sets the height.
		 * @param height The height to set
		 */
		public void setHeight(int height) {
			this.height = height;
		}

		/**
		 * Sets the width.
		 * @param width The width to set
		 */
		public void setWidth(int width) {
			this.width = width;
		}

		/**
		 * Returns the filename.
		 * @return String
		 */
		public String getFilename() {
			return filename;
		}

		/**
		 * Sets the filename.
		 * @param filename The filename to set
		 */
		public void setFilename(String filename) {
			this.filename = filename;
		}


		/**
		 * Returns the max_min_Dn.
		 * @return float[][]
		 * // Retrieve both the maximum and minimum pixel value
		 */
		public double[][] getMax_min_Dn() {
			// Set up the parameter block for the source image and
		 // the constants
		//ROI roi = new ROI(loadImage);

		ParameterBlock pb = new ParameterBlock();
		pb.addSource(loadImage);   // The source image
		//pb.add(null);        // The region of the image to scan
		//pb.add(1);         // The horizontal sampling rate
		//pb.add(1);         // The vertical sampling rate

		// Perform the extrema operation on the source image
		RenderedOp op = JAI.create("extrema", pb);


			max_min_Dn =  (double[][]) op.getProperty("extrema");

			return max_min_Dn;
		}

		/*
		 * write one pixel in location (x, y)
		 */
		public void setMapValue(int x, int y, int dn) {
			WritableRandomIter witer = (WritableRan-
domIter)RandomIterFactory.createWritable(blankImage, null);

			for(int i =0; i < width; i++){
				//float dn = iter.getSampleFloat(i, 150, 0);
				witer.setSample(x,y,0,dn);
			}

		}

		public void writeFile(String outputFile) {

			// Encode the file as a BMP image.
			FileOutputStream stream;
			try {

				stream = new FileOutputStream(outputFile + "." + fileFormat);
				// Set the encoding parameters if necessary.
				JPEGEncodeParam encodeParam = null;

				encodeParam = new JPEGEncodeParam();
```

```
                              encodeParam.setQuality(1.0F);

                              ImageEncoder enc = ImageCodec.createImageEncoder(fileFormat,
stream,encodeParam);
                              try {
                                     enc.encode(blankImage);
                              } catch (IOException e) {
                                     System.out.println("又完蛋啦! \n");
                              }

                              JAI.create("encode", blankImage, stream, fileFormat,null);


                       } catch (FileNotFoundException e) {
                              System.out.println("can not open output file\n");
                       }

                       // Store the image in the BMP format. mapbe this call is the same as
"encode" one
                       //JAI.create("filestore", blankImage, outputFile, fileFormat, null);

               }

}
```

- Class LandUse

```
/*
 * (c) 2002 OCT The ITC Enchede, All Rights Reserved
 *
 * Permission to use this software and its documentation for non-commercial
 * purposes and without fee is hereby granted, provided this copyright statement
 * is included. Please contact us for permission for redistribution and other uses.
 *
 * this CA model used to simulate the urban--non-urban development process in ZS
 * while the pakage name landuse is inherated from early consideration of landuse change simu-
lating
 *
 * Version 0.4
 */

package zssim;

import java.awt.*;
import java.awt.image.*;
import java.io.FileNotFoundException;
import java.util.*;

import edu.brook.ascape.model.*;
import edu.brook.ascape.rule.*;

import edu.brook.ascape.util.*;
import edu.brook.ascape.view.*;



/**
 * my simple test of CA to simulate the land use change using very simple rules
 */

public class LandUse extends ScapeVector {

        /*
         * this part declare the parameters to control the model
         */

        protected static final int height = 376, width = 606;
        protected static final int totalNumberOfCells = height * width;

        protected boolean compensative = true;
        protected int demands = 459;
        protected int alpha = 1; // the stochastic disturbance defree
        protected int neighborhoodSize = 2; //the search distance actually
        protected float weightMainCenter = 0.5f, weightSubCenter = 0.5f;
        protected float weight1ClassRoad = 0.7f, weight2ClassRoad = 0.3f;
        protected float betaForUrbanForm = 0.5f,betaForRoadnet = 0.5f;

        //weight system for neighbourhood, urban form, tranport net and terrain
        //here to untilize the setting window, the value has to be divided by 100
        protected int weight_Neighbourhood = 20, weight_urbanForm = 20, weight_Roadnet = 30,
weight_terrain = 20,weigh_Masterplan = 10;

        protected boolean turnOnCharts = false;
```

```
        //parameter sweep ?????
        protected boolean doSweep = false;

        //save the statistic data?
        protected boolean saveStatData = false;


        /*
         * here is the variables to be used in the model
         */
        ImageContainer myImage;
        String workspace = "d:\\MSC_thesis\\data\\CA\\intermedia\\";

        //use as a container to sort the top most
        protected static float[] potentials = new float[height*width];
        //an index cursor indicate the current location of pointer, default is 0
        protected static int posOfPotentials = 0;
        protected static int iterationCounter = 0;
        private float threshold = 0;

        /*
         * facility functions
         */
         /*
          * save result to tif file
          */
        protected void saveResult(){



        }
        /**
         * Import map and water sources from binary data files.
         */
        public void importMap(int width, int height) {


                final String[] ImageList = {
                        "lu80_with_subcenter.tif",
                        "road net\\dist2rd1_80_score.tif",
                        "water body\\forbidden.tif",
                        //"dem\\dem_idw.tif",
                        "dem\\slop_percent.tif",
                        "urban form\\attractive_e_0.5_of_twocenters.tif",
                        "urban form\\dist2main_center.tif",
                        "urban form\\dist2sub_center.tif",
                        "road net\\dist2rd1_80.tif",
                        "road net\\dist2rd2_80.tif"
                        };

                //first image of Landuse 80
                myImage = new ImageContainer(workspace + ImageList[0]);
                System.out.println("height:" + myImage.getHeight() + "\twidth:" + myI-
mage.getWidth() + "\n");

                //i do not need many bands now
                //int bands = loadImage.getSampleModel().getNumBands();

                //for (int band=0; band < bands; band++) {
                for (int line = 0; line < height; line++) {
                        for (int samp = 0; samp < width; samp++) {
                                //int dn = iter.getSample(samp, line, 0);
                                int dn = myImage.getMapValue(samp, line); // whatever

                                landCell location = (landCell) ((ScapeArray2DMoore) lat-
tice).getCell(samp,line);
                                location.setLandtype((byte) dn);
                        }
                }
                //}

                System.out.println("finish loading the image\n"      + "width:" + width + "
height:" + height);

                //the second is accessibility --the distance decay score to the main road in
the area
                myImage = new ImageContainer(workspace + ImageList[1]);

                //get the max min DN value of the image to scale them into scores
                /*
                *since the exponent function automaticly normalized the dn value, this
                *part is not needed anymore
                *
                double[][] extrema = myImage.getMax_min_Dn();
                double minDn = extrema[0][0];
                double maxDn = extrema[1][0];
```

```
		*/
		for (int row = 0; row < height; row++) {
			for (int column = 0; column < width; column++) {

				float dn = myImage.getfMapValue(column, row); // whatever

				landCell location =
					(landCell) ((ScapeArray2DMoore) lat-
tice).getCell(column,row);

				location.setAccessibility(dn);
			}
		}

		System.out.println("accessibility map's height:" + myImage.getHeight() + "\t
and \twidth:" + myImage.getWidth()    + "\n");

		//now is the water  body
		//the water body almost keep no change during the last two decades
		myImage = new ImageContainer(workspace + ImageList[2]);

		for (int row = 0; row < height; row++) {
			for (int column = 0; column < width; column++) {

				int dn = myImage.getMapValue(column, row); // whatever

				landCell location =
					(landCell) ((ScapeArray2DMoore) lattice).getCell(
						column,
						row);
				location.setStreams((int) dn > 0 ? 1 : 0);
			}
		}

		System.out.println("water body map's height:"+ myImage.getHeight() + "\t and
\twidth:" + myImage.getWidth()        + "\n");

		//the second is accessibility --the distance decay score to the main road in
the area
		myImage = new ImageContainer(workspace + ImageList[3]);

		for (int row = 0; row < height; row++) {
			for (int column = 0; column < width; column++) {

				float dn = myImage.getfMapValue(column, row); // whatever

				landCell location = (landCell) ((ScapeArray2DMoore) lat-
tice).getCell(column,row);

				location.setDem(dn);
			}
		}

		System.out.println("DEM map's height:" + myImage.getHeight() + "\t and
\twidth:" + myImage.getWidth()        + "\n");

		//the effect of urban centers
		myImage = new ImageContainer(workspace + ImageList[4]);

		for (int row = 0; row < height; row++) {
			for (int column = 0; column < width; column++) {

				float dn = myImage.getfMapValue(column, row); // whatever

				landCell location =    (landCell) ((ScapeArray2DMoore) lat-
tice).getCell(column,row);

				location.setAttractiveness(dn);
			}
		}

		System.out.println("centrun effect map's height:" + myImage.getHeight() + "\t
and \twidth:" + myImage.getWidth()    + "\n");

		//read distance to main center
		myImage = new ImageContainer(workspace + ImageList[5]);

		for (int row = 0; row < height; row++) {
			for (int column = 0; column < width; column++) {

				float dn = myImage.getfMapValue(column, row); // whatever

				landCell location = (landCell) ((ScapeArray2DMoore) lat-
tice).getCell(column,row);

				location.setDist2maincenter(dn);
			}
```

```
                }

                System.out.println("centrun effect map's height:" + myImage.getHeight() + "\t
and \twidth:" + myImage.getWidth()    + "\n");

                //read distance to sub center
                myImage = new ImageContainer(workspace + ImageList[6]);

                for (int row = 0; row < height; row++) {
                        for (int column = 0; column < width; column++) {

                                float dn = myImage.getfMapValue(column, row); // whatever

                                landCell location = (landCell) ((ScapeArray2DMoore) lat-
tice).getCell(column,row);

                                location.setDist2subcenter(dn);
                        }
                }

                System.out.println("centrun effect map's height:" + myImage.getHeight() + "\t
and \twidth:" + myImage.getWidth()    + "\n");

                //the distance to main road
                myImage = new ImageContainer(workspace + ImageList[7]);

                for (int row = 0; row < height; row++) {
                        for (int column = 0; column < width; column++) {

                                float dn = myImage.getfMapValue(column, row); // whatever

                                landCell location =    (landCell) ((ScapeArray2DMoore) lat-
tice).getCell(column,row);

                                location.setDist2road1(dn);
                        }
                }
                System.out.println("distance to main road loaded\n");

                //the distance to 2nd class road
                myImage = new ImageContainer(workspace + ImageList[8]);

                for (int row = 0; row < height; row++) {
                        for (int column = 0; column < width; column++) {

                                float dn = myImage.getfMapValue(column, row); // whatever

                                landCell location =    (landCell) ((ScapeArray2DMoore) lat-
tice).getCell(column,row);

                                location.setDist2road2(dn);
                        }
                }
                System.out.println("distance to second class road loaded\n");

        }

        /**
         * Import map and water sources from binary data files.
         */
        public void importRoad92(int width, int height) {

                final String[] ImageList = {
                        "road net\\dist2rd1_92.tif",
                        "road net\\dist2rd1_92.tif"
                };

                //the distance to main road
                myImage = new ImageContainer(workspace + ImageList[0]);

                for (int row = 0; row < height; row++) {
                        for (int column = 0; column < width; column++) {

                                float dn = myImage.getfMapValue(column, row); // whatever

                                landCell location =    (landCell) ((ScapeArray2DMoore) lat-
tice).getCell(column,row);

                                location.setDist2road1(dn);
                        }
                }
                System.out.println("92 distance to main road loaded\n");

                //the distance to 2nd class road
                myImage = new ImageContainer(workspace + ImageList[1]);

                for (int row = 0; row < height; row++) {
```

```
                        for (int column = 0; column < width; column++) {

                                float dn = myImage.getfMapValue(column, row); // whatever

                                landCell location =    (landCell) ((ScapeArray2DMoore) lat-
tice).getCell(column,row);

                                location.setDist2road2(dn);
                        }
                }
                System.out.println("92 distance to second class road loaded\n");

        }

        /*
         * write out the urban map
         */
        public void exportMap(String filename) {
                //set the dimension of the map file
                ImageContainer image = new ImageContainer(width,height); //set a buffered image
dimesions

                //wtite to temporal image pixel by pixel
                for (int row = 0; row < height; row++) {
                        for (int column = 0; column < width; column++) {

                                landCell location =    (landCell) ((ScapeArray2DMoore) lat-
tice).getCell(column,row);

                                int dn = (int)(location.getLandtype());
                                image.setMapValue(column, row,dn*255); // whatever

                        }
                }

                //write out
                image.writeFile(filename);

        }

        /*
         * facility functions
         */
        protected void addPotentials(float _potential){
                potentials[posOfPotentials] = _potential;
                if(_potential > 1) {
                        System.out.println("\t\"" + _potential);
                }
                posOfPotentials++;

                //System.out.println(posOfPotentials + "\n");

                if(posOfPotentials >= totalNumberOfCells) {//which means all cells have been
processed

                        Arrays.sort(potentials); //to be used later on
                        posOfPotentials = 0; //set to zero
                        threshold = potentials[totalNumberOfCells -1 - demands];

                        /*
                         * to test the iteration
                         */
                        System.out.println("################\n" + potentials[totalNumberOfCells
-1]);

                        Arrays.fill(potentials,0f);

                        iterationCounter++;

                        //here to be revised ,maybe parameters sweep can do something
                        if(iterationCounter == 11){//reach 1992
                                setPaused(true); //actually both call to pause call the same
method executeOnRoot(PAULSE_RULE), no difference
                                //pause();  //to capture the image, should be replaced by image
save function later on
                                importRoad92(width,height); //change road net
                                setDemands(946);
                        }

                        //String filename = workspace + "output data\\urban" + iterationCounter;

                        //exportMap(filename); //export for each iteration
                }
        }


        //ScapeArray2DVonNeumann lattice;
```

```
	ScapeGraph lattice;

	public void createScape() {

		super.createScape();

		setAutoRestart(false);

		//startOnOpen = false; //take some time to find this instruction.
		setStartOnOpen(false);
		stopPeriod = 20;



		//Create a 2D lattice (we'll be using von Neumann Geometry.)
		lattice = new ScapeArray2DMoore();

		lattice.setPrototypeAgent(new landCell());
		lattice.setAutoCreate(false);


		//lattice.getRules().clear();//  this clear the default iteration for the back-
groud cells
		//exactly it is! "iterate"

		//lattice.addRule(NEXT_STATE_SYNCHRONOUS); i add the rules to landCell  in its
class definition
		lattice.setExecutionOrder(RULE_ORDER);

		//set the extent to the common boundary

		lattice.setExtent(new Coordinate2DDiscrete(width, height));

		//set the space as aperiodic space
		lattice.getGeometry().setPeriodic(false);

		((ScapeArray2D)lattice).setPreSetNeighbors(false);

		lattice.createScape();

		importMap(width, height);

		addAgent(lattice);

	}

	public void createViews() {
		super.createViews();

		/*
		 * this part use to add those statistical functions
		 */
		final StatCollector[] statistics = {
			/*
			new StatCollectorCond("transited") {
				public boolean meetsCondition(Object object) {
					return (((landCell)object).getAttractiveness());
				}
			}
			*/
			new StatCollector("Rank_top10") {
				public double getValue(Object object) {
					return ((landCell)object).getPotential();
				}
				public String getName() {
					return "potential";
				}
			}
		};

		lattice.addStatCollectors(statistics);


		Overhead2DView view = new Overhead2DView();
		view.setCellSize(1);
		lattice.addView(view);

		/*
		 * chart view
		 */
		if(turnOnCharts){
			ChartView totalTransitChart = new ChartView();
			lattice.addView(totalTransitChart);
			totalTransitChart.addSeries( "test",Color.RED); //to be think carefully
later on
		}
```

```
		/*
		 * parameter sweep  ???? everybody says it is usefull, not really
		 * understand
		 */
		if(doSweep) {  //default is false
			SweepControlView sweeper = new SweepControlView();
			lattice.addView(sweeper);

			SweepDimension test1 = new SweepDimension((LandUse) get-
Model(),"test1",1,5,1);

			SweepGroup toSweep = new SweepGroup();
			toSweep.addMember( test1);

			sweeper.setSweepGroup(toSweep);
			toSweep.setRunsPer(1);
		} //total not understand  ????????

	}



	/**
	 * Returns the threshold.
	 * @return float
	 */
	public float getThreshold() {
		return threshold;
	}

	/**
	 * Sets the threshold.
	 * @param threshold The threshold to set
	 */
	public void setThreshold(float threshold) {
		this.threshold = threshold;
	}

	/**
	 * Returns the demands.
	 * @return int
	 */
	public int getDemands() {
		return demands;
	}

	/**
	 * Sets the demands.
	 * @param demands The demands to set
	 */
	public void setDemands(int demands) {
		this.demands = demands;
	}

	/**
	 * Returns the alpha.
	 * @return int
	 */
	public int getAlpha() {
		return alpha;
	}

	/**
	 * Sets the alpha.
	 * @param alpha The alpha to set
	 */
	public void setAlpha(int alpha) {
		this.alpha = alpha;
	}

	/**
	 * Returns the neighborhoodSize.
	 * @return int
	 */
	public int getNeighborhoodSize() {
		return neighborhoodSize;
	}

	/**
	 * Sets the neighborhoodSize.
	 * @param neighborhoodSize The neighborhoodSize to set
	 */
	public void setNeighborhoodSize(int neighborhoodSize) {
		this.neighborhoodSize = neighborhoodSize;
	}
```

```
/**
 * Returns the weightMainCenter.
 * @return float
 */
public float getWeightMainCenter() {
        return weightMainCenter;
}

/**
 * Sets the weightMainCenter.
 * @param weightMainCenter The weightMainCenter to set
 */
public void setWeightMainCenter(float weightMainCenter) {
        this.weightMainCenter = weightMainCenter;
}

/**
 * Returns the weightSubCenter.
 * @return float
 */
public float getWeightSubCenter() {
        return weightSubCenter;
}

/**
 * Sets the weightSubCenter.
 * @param weightSubCenter The weightSubCenter to set
 */
public void setWeightSubCenter(float weightSubCenter) {
        this.weightSubCenter = weightSubCenter;
}

/**
 * Returns the betaForRoadnet.
 * @return float
 */
public float getBetaForRoadnet() {
        return betaForRoadnet;
}

/**
 * Sets the betaForRoadnet.
 * @param betaForRoadnet The betaForRoadnet to set
 */
public void setBetaForRoadnet(float betaForRoadnet) {
        this.betaForRoadnet = betaForRoadnet;
}

/**
 * Returns the betaForUrbanForm.
 * @return float
 */
public float getBetaForUrbanForm() {
        return betaForUrbanForm;
}

/**
 * Sets the betaForUrbanForm.
 * @param betaForUrbanForm The betaForUrbanForm to set
 */
public void setBetaForUrbanForm(float betaForUrbanForm) {
        this.betaForUrbanForm = betaForUrbanForm;
}



/**
 * Returns the compensative.
 * @return boolean
 */
public boolean isCompensative() {
        return compensative;
}

/**
 * Sets the compensative.
 * @param compensative The compensative to set
 */
public void setCompensative(boolean compensative) {
        this.compensative = compensative;
}

/**
 * Returns the weight1ClassRoad.
 * @return float
 */
public float getWeight1ClassRoad() {
```

```
        return weight1ClassRoad;
}

/**
 * Returns the weight2ClassRoad.
 * @return float
 */
public float getWeight2ClassRoad() {
        return weight2ClassRoad;
}

/**
 * Sets the weight1ClassRoad.
 * @param weight1ClassRoad The weight1ClassRoad to set
 */
public void setWeight1ClassRoad(float weight1ClassRoad) {
        this.weight1ClassRoad = weight1ClassRoad;
}

/**
 * Sets the weight2ClassRoad.
 * @param weight2ClassRoad The weight2ClassRoad to set
 */
public void setWeight2ClassRoad(float weight2ClassRoad) {
        this.weight2ClassRoad = weight2ClassRoad;
}

/**
 * Returns the workSpace.
 * @return String
 */
public String getWorkSpace() {
        return workspace;
}

/**
 * Sets the workSpace.
 * @param workSpace The workSpace to set
 */
public void setWorkSpace(String workSpace) {
        this.workspace = workSpace;
}

/**
 * Returns the weigh_Neighbourhood.
 * @return int
 */
public int getWeigh_Neighbourhood() {
        return weight_Neighbourhood;
}

/**
 * Returns the weight_Roadnet.
 * @return int
 */
public int getWeight_Roadnet() {
        return weight_Roadnet;
}

/**
 * Returns the weight_terrain.
 * @return int
 */
public int getWeight_terrain() {
        return weight_terrain;
}

/**
 * Returns the weight_urbanForm.
 * @return int
 */
public int getWeight_urbanForm() {
        return weight_urbanForm;
}

/**
 * Returns the workspace.
 * @return String
 */
public String getWorkspace() {
        return workspace;
}

/**
 * Sets the weigh_Neighbourhood.
 * @param weigh_Neighbourhood The weigh_Neighbourhood to set
 */
```

```
	public void setWeigh_Neighbourhood(int weigh_Neighbourhood) {
		this.weight_Neighbourhood = weigh_Neighbourhood;
	}

	/**
	 * Sets the weight_Roadnet.
	 * @param weight_Roadnet The weight_Roadnet to set
	 */
	public void setWeight_Roadnet(int weight_Roadnet) {
		this.weight_Roadnet = weight_Roadnet;
	}

	/**
	 * Sets the weight_terrain.
	 * @param weight_terrain The weight_terrain to set
	 */
	public void setWeight_terrain(int weight_terrain) {
		this.weight_terrain = weight_terrain;
	}

	/**
	 * Sets the weight_urbanForm.
	 * @param weight_urbanForm The weight_urbanForm to set
	 */
	public void setWeight_urbanForm(int weight_urbanForm) {
		this.weight_urbanForm = weight_urbanForm;
	}

	/**
	 * Sets the workspace.
	 * @param workspace The workspace to set
	 */
	public void setWorkspace(String workspace) {
		this.workspace = workspace;
	}

	/**
	 * Returns the weigh_Masterplan.
	 * @return int
	 */
	public int getWeigh_Masterplan() {
		return weigh_Masterplan;
	}

	/**
	 * Sets the weigh_Masterplan.
	 * @param weigh_Masterplan The weigh_Masterplan to set
	 */
	public void setWeigh_Masterplan(int weigh_Masterplan) {
		this.weigh_Masterplan = weigh_Masterplan;
	}

}
```

- Class landCell

```
/*
 * (c) 2002 OCT The ITC Enchede, All Rights Reserved
 *
 * Permission to use this software and its documentation for non-commercial
 * purposes and without fee is hereby granted, provided this copyright statement
 * is included. Please contact us for permission for redistribution and other uses.
 *
 * this CA model used to simulate the urban--non-urban development process in ZS
 * while the pakage name landuse is inherated from early consideration of landuse change simu-
lating
 *
 * Version 0.4
 */

package zssim;

import java.awt.*;
import java.util.*;

import edu.brook.ascape.model.*;
import edu.brook.ascape.rule.*;

import edu.brook.ascape.util.*;



/**
 * @author szw
 *
 * To change this generated comment edit the template variable "typecomment":
```

```
 * Window>Preferences>Java>Templates.
 * To enable and disable the creation of type comments go to
 * Window>Preferences>Java>Code Generation.
 */
/**
 * This is the class that defines the state of a cell within the automata.
 */
public class landCell extends Cell {


        /**
         * this rule is not used in this version
                */
        public final static Rule NEXT_STATE_SYNCHRONOUS =
                new ExecuteThenUpdate("Gray Cell approach") {
                public void execute(Agent agent) {
                        ((landCell) agent).setPotential(((landCell)
agent).calculatePotential_graycell() );

                }
                public void update(Agent agent) {
                        if(((landCell) agent).getPotential() >= 0.5){  //ready to develop into
urban
                                ((landCell) agent).landtype = 1;
                                ((landCell)agent).setPotential(0f);
                        }
                }
        };

        //another rule to calculate the potential developing value
        public final static Rule DEVELOP_POTENTIAL =
                new Rule("calculate the potentials") {
                public void execute(Agent agent) {
                        //calculate potentials
                        ((landCell) agent).setPotential( ((landCell)
agent).calculatePotential());
                        //updates array of potentials used to be sorted for getting the top most
ones to develop later on

        //((LandUse)(scape.getModel())).addPotentials(((landCell)agent).getPotential());
                        (((Lan-
dUse)agent.getRoot())).addPotentials(((landCell)agent).getPotential());
                }
                public boolean isRandomExecution() {
                        return false;
                }
                public boolean isCauseRemoval() {
                        return false;
                }

        };

        private byte landtype; //this time only 0-undeveloped and 1-developed
        private int streams;
        private float accessibility;
        //maybe need to be change later, should be scaled to 0--1 range
        private float attractiveness; //the effect of urban centers
        private float potential; // the potential of transition
        private float dem;  //terrain
        private float slope; // slope

        private float dist2maincenter;
        private float dist2subcenter;
        private float dist2road1,dist2road2;

        public void scapeCreated() {
                scape.addRule(DEVELOP_POTENTIAL);

                scape.addRule(UPDATE_RULE);
                //scape.addRule(NEXT_STATE_SYNCHRONOUS);

        }

        /**
         * to set a filter to get those who meets the conditions
         */
        private final static Conditional CON_DEVELOPED = new Conditional() {
                public boolean meetsCondition(Object object) {
                        return ((landCell) object).landtype == 1;

                }
        };

        /*
         * conditional to filter out those lower potential values--maybe usefull
         * later
         */
```

```
        private final static Conditional TOPMOSTCELLS = new Conditional() {
                public boolean meetsCondition(Object object) {


                        return ((landCell) object).potential > ((Lan-
dUse)((landCell)object).getScape().getModel()).getThreshold();
                }
        };


        //to initialize the model , namely use "0" to fill all the cells
        /*
        public void initialize() {
            super.initialize();
                //maybe something to be done to initialize the lattice when start
                if (getRandom().nextFloat() < 0.1){
                        landtype = 1;
                }
        }*/

        //calculate the potential value for each cell, mutual exclusive with calculate poten-
tials rule
        public float calculatePotential_graycell() { //this is something like utility scores
                float pValue = this.potential;
                if (this.landtype == 1) { //if it was developed, just continue the process
                        return 0f;
                }
                else {
                        pValue = calculatePotential() + this.potential;
                        return pValue;
                }
        }

                //calculate the potential value for each cell, this time no cumulative process
        public float calculatePotential() {
                //float pValue = this.potential;
                double pValue;
                if (this.landtype == 1) { //if it was developed, just continue the process and
return 0
                        return 0f;
                }
                else {
                        //this neighborsDEV_num use the default 3X3 moore neighborhood
                        /*  this part use the default neighborhood defination, have to use
                         *  alternatives before i can redefine the neighborhood with
                         *  different size while not affecting the performance
                         */

                        /*
                         * this two functions have to be compared later
                         * it seems that countNeighbor is different from countWithin
                         */

                        /*this one does not implement the Moore neighbor inquiry method
                        int neighborsDEV_num =
this.countWithin(CON_DEVELOPED,true,((LandUse)getScape().getModel()).getNeighborhoodSize());
                        */

                        /*
                         * test an alternative to count in geometry specified space
                         */
                        int neighborhoodSize = ((Lan-
dUse)getScape().getModel()).getNeighborhoodSize();
                        int neighborsDEV_num =
my_countWithinMoore(CON_DEVELOPED,true,neighborhoodSize);


                        //add stochastic disturbance  alpha to control the dispersal degree
                        double alpha = ((LandUse)getScape().getModel()).getAlpha();
                        Random myrandom = new Random();
                        double stochastic = 1 + Math.pow(-
Math.log(myrandom.nextDouble()),alpha);

                        double terrainConstraint = getDem() > 100 ? 0f : (1f - getDem() / 100f);
//Math.exp(-0.5 * getDem());
                        double urbanFormEffect = cal_form_effect();

                        //gradience of impedance doutble work in non-compensatory way has no
meaning
                        double K = 1;
                        int xSize = ((ScapeArray2D)(getScape())).getXSize();
                        double impedance = K * ((Coordi-
nate2DDiscrete)getCoordinate()).getXValue() / xSize;//((LandUse)getScape().getModel()).getK();

                        double accessibility = cal_road_attract();
```

```
                        if(!(((LandUse)getScape().getModel()).isCompensative())){ //un-
compensative
                                pValue = stochastic * ((double)neighborsDEV_num / 22500d) * ur-
banFormEffect * (double)((1 - this.streams)) * (double)accessibility * terrainConstraint *
impedance;

                        }
                        else {//compensative
                                /*
                                 * first of all, get the weight system
                                 */
                                double w1,w2,w3,w4,w5;
                                //w1 --> neighborhood, w2 --> urban form, w3 --> tranport net,
w4 --> terrain
                                w1 = ((LandUse)getScape().getModel()).getWeigh_Neighbourhood() /
100f;
                                w2 = ((LandUse)getScape().getModel()).getWeight_urbanForm() /
100f;
                                w3 = ((LandUse)getScape().getModel()).getWeight_Roadnet() /
100f;
                                w4 = ((LandUse)getScape().getModel()).getWeight_terrain() /
100f;
                                w5 = ((LandUse)getScape().getModel()).getWeigh_Masterplan() /
100f;

                                pValue = /*stochastic */ (double)(this.streams > 0 ? 0 :1 ) * (
((double)neighborsDEV_num / ((neighborhoodSize * 2 +1) * (neighborhoodSize * 2 + 1)) ) * w1 +
urbanFormEffect * w2 +  (double)accessibility * w3 + terrainConstraint * w4 + impedance * w5);

                        }

                        //System.out.println("accessibility:" + accessibility + "\tpotential:" +
potential + "\tstreams: " + streams + "\n");
                        if(pValue > 1) {
                                System.out.print(pValue);
                        }

                        return (float)pValue;
                }
        }
        /**
         * Method my_countWithinMoore.
         * @param CON_DEVELOPED
         * @param b
         * @param i
         */
        private int my_countWithinMoore(Conditional condition,boolean includeOrigin,int search-
Distance) {
                int counter = 0;

                Cell[] neighbors= this.getCellsNear(searchDistance,true);

                for(int i = 0; i < neighbors.length; i++){
                        if(condition.meetsCondition(neighbors[i])){
                                counter++;
                        }

                }
                if(!includeOrigin){
                        counter--;
                }
                return counter;
        }

        /*
         * update the cell states according to the threshold
         */
        public void update() {
                //convert those cells into urban land type if they have higher values than
threshold
                if(getPotential() > ((LandUse)(getScape().getModel())).getThreshold())
                {
                        landtype = 1;
                }
        }

        //to be normalized
        private double cal_form_effect(){
                //beta is the parameter used to control the decay
                double beta = ((LandUse)getScape().getModel()).getBetaForUrbanForm();
                double attract = (dou-
ble)(((LandUse)(getScape().getModel())).getWeightMainCenter()) * Math.exp(-1 * (double)beta *
dist2maincenter / 1000 ) +
                        (double)(((LandUse)(getScape().getModel())).getWeightSubCenter()) *
Math.exp(-1 * (double)beta * dist2subcenter / 1000);

                return (float) attract;
```

```
        }

        //to be normalized  !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!1
        private double cal_road_attract(){
                //beta is the parameter used to control the decay rate
                double beta = 0.5;
                double attract = (dou-
ble)(((LandUse)(getScape().getModel())).getWeight1ClassRoad()) * Math.exp(-1 * (double)beta *
dist2road1 / 1000 ) +
                        (double)(((LandUse)(getScape().getModel())).getWeight2ClassRoad()) *
Math.exp(-1 * (double)beta * dist2road2 / 1000);

                        return (float) attract;

        }

        public Color getColor() {
                Color rcolor;

                //if (dist2road1 > 100 || dist2road2 > 100 ){

                                        switch (landtype) {
                                                case 0 :
                                                        rcolor = Color.white;
                                                        break;
                                                case 1 :
                                                        rcolor = Color.red;
                                                        break;
                                                default :
                                                        rcolor = Color.BLACK;
                                                        break;
                                        }
                //}
                /*
                else {
                        rcolor = Color.black;
                }
                */
                return rcolor;
        }
        /**
         * Returns the landtype.
         * @return short
         */
        public byte getLandtype() {
                return landtype;
        }

        /**
         * Sets the landtype.
         * @param landtype The landtype to set
         */
        public void setLandtype(byte landtype) {
                this.landtype = (byte) (landtype);
        }

        /**
         * Returns the streams.
         * @return byte
         */
        public int getStreams() {
                return streams;
        }

        /**
         * Sets the streams.
         * @param streams The streams to set
         */
        public void setStreams(int streams) {
                this.streams = streams;
        }

        /**
         * Returns the potential.
         * @return float
         */
        public float getPotential() {
                return potential;
        }

        /**
         * Sets the potential.
         * @param potential The potential to set
         */
        public void setPotential(float potential) {
                this.potential = potential;
```

```
}

/**
 * Returns the dem.
 * @return float
 */
public float getDem() {
        return dem;
}

/**
 * Sets the dem.
 * @param dem The dem to set
 */
public void setDem(float dem) {
        this.dem = dem;
}

/**
 * Returns the dist2maincenter.
 * @return float
 */
public float getDist2maincenter() {
        return dist2maincenter;
}

/**
 * Sets the dist2maincenter.
 * @param dist2maincenter The dist2maincenter to set
 */
public void setDist2maincenter(float dist2maincenter) {
        this.dist2maincenter = dist2maincenter;
}

/**
 * Returns the dist2subcenter.
 * @return float
 */
public float getDist2subcenter() {
        return dist2subcenter;
}

/**
 * Sets the dist2subcenter.
 * @param dist2subcenter The dist2subcenter to set
 */
public void setDist2subcenter(float dist2subcenter) {
        this.dist2subcenter = dist2subcenter;
}

/**
 * Returns the attractiveness.
 * @return short
 */
public float getAttractiveness() {
        return attractiveness;
}

/**
 * Sets the attractiveness.
 * @param attractiveness The attractiveness to set
 */
public void setAttractiveness(float attractiveness) {
        this.attractiveness = attractiveness;
}

/**
 * Returns the accessibility.
 * @return float
 */
public float getAccessibility() {
        return accessibility;
}

/**
 * Sets the accessibility.
 * @param accessibility The accessibility to set
 */
public void setAccessibility(float accessibility) {
        this.accessibility = accessibility;
}

/**
 * Returns the dist2road1.
 * @return float
 */
public float getDist2road1() {
```

```
		return dist2road1;
	}

	/**
	 * Returns the dist2road2.
	 * @return float
	 */
	public float getDist2road2() {
		return dist2road2;
	}

	/**
	 * Sets the dist2road1.
	 * @param dist2road1 The dist2road1 to set
	 */
	public void setDist2road1(float dist2road1) {
		this.dist2road1 = dist2road1;
	}

	/**
	 * Sets the dist2road2.
	 * @param dist2road2 The dist2road2 to set
	 */
	public void setDist2road2(float dist2road2) {
		this.dist2road2 = dist2road2;
	}

}
```