CA and MAS have been used in the past to model urban systems but they both have certain limitations.

Weaknesses of CA are:
1. Regular partitioning of space when most urban phenomenon is irregularly partitioned.
2. Inability of cells to move about.

“The CA framework is regarded as insufficient in dealing with mobile objects such as pedestrians, migrating households, or relocating firms.

This immobility issue has led to increased research interest in MAS. The main geographic advantage of agent automata lies in their ability to transmit information by themselves, moving to another location, which can be at any distance from an agent’s current position.

Weakness of MAS:
1. Most efforts to model behavior with a multivalent system outside geography have been non-spatial.

“the amalgamation of CA and MAS tools for urban simulation necessitates certain awkward methodological compromises and most combined CA-MAS computer environments and application exploit a strict CA view of the geographic systems the they model. CA cells are granted some degree of agency in their state descriptions and are simply reinterpreted as artificial agents (Box, 2001) and/or MAS are imposed on top of CA and simulated agents are interpreted as responding to averaged cell conditions.”

In this amalgamation there are certain elements which are essential to simulating an urban system:

1. A typology of entities
2. The space in which they are situated
3. The spatial relationships between entities
4. The processes governing the changes of entities’ characteristics
5. The processes governing the changes of their location in space.
Neither CA nor MAS can fully provide these requirements in isolation. The geography of the CA framework is problematic for urban simulation because CA is incapable of representing autonomously mobile entities. MAS are weak as a single tool because of the generality of the concept and the broad problem that existing MAS tools and methodologies still underestimate the importance of space and relocation behavior. In many cases, the simulated entities represented by CA and MAS models do not behave as we understand they should, largely because the modeling framework will not permit them to.

The Geographic Automata System

Geographic Automata systems (GAS)

The Geographic Automata System (GAS) framework unites CA and MAS formalism in such a way as to directly reflect a geographic and object-based-more specifically, automata-based view of urban systems. However, this necessitates a re-working of the automata ideas to incorporate the five essential components mentioned in Section 1 of this chapter.

In general, automata are characterized by states and transitions rules. In the case of geographic automata, we also introduce functionality to enable the explicit consideration of space and spatial behavior. In CA, pre-defined partitions are often used as a proxy for geography.

With GAS, instead of predefined partitions, we introduce an independent set of georeferencing rules for situating geographic automata in space. Likewise, we define neighborhood rules, rather than relying on fixed neighborhood patterns that are incapable of being varied in space or time once delineated.

To incorporate the mobility of MAS, “we also consider a set of independent movement rules that allow for the independent navigation of geographic automata in their simulated environments.

Formally, a Geographic Automata System (GAS), G, may be defined as consisting of seven components:

\[ G = (K; S; \overline{S}; L; M; \overline{L}; N; R_N) \]

- \( K \) denotes a set of types of automatons featured in the given GAS
- \( \overline{S} \) is the set of state that the different automatons can take on
- \( \overline{T} \) is the transition rules used to determine how automatons states should change over time.
\( L \) denotes the georeferencing conventions that dictate the location of automata in the system

\( M \) denotes the movement rules for automata

\( N \) represents the neighbors of the automata

\( R \) represents the neighborhood rules that govern how automata relate to one another.

If we go back to the list of necessary qualities for geosimulation we see that they are contained in the above elements.

1. A typology of entities
2. The space in which they are situated
   1. The spatial relationships between entities
   2. The processes governing the changes of entities’ characteristics
   3. The processes governing the changes of their location in space.

Geographic Automata Types

1. Fixed vs. Nonfixed
   Building footprints, road networks, parks, vs. pedestrians, vehicles, householders.

Any urban system obviously contains both fixed and no fixed entities. All entities have associated characteristics:
For fixed entities there could be attributes such as number of rooms, value, number of floors, architectural style, etc.

Non-fixed entities can have attributes such as economic status, number of children, mean age of parents. The characteristics of non-fixed automata often depend on each other. For example, the value of an apartment depends on the real estate in the property and property’s neighborhood and on the population of the neighborhood.

Geographic Automata states and transition rules.

Any variable can be used to derive state values, including variables of geographic significance height, accessibility, visibility, etc. In the case of non-fixed automata, state variables of relevance to the movement rules of the system may be introduced, for example, heading, speed, progress, toward destination.

Transition rules could be things like real estate value changing based on the states of the real estate objects and, more importantly, on various attributes of the households that occupy them. In terms of (nonfixed) householder geographic automata, a transition rule describing changes in the economic status of a householder could be defined. Similarly, we could specify a rule, describing the way households change their residence, and describing how householder’s neighbors are determined.

Geographic Automata spatial referencing and migration rules.
For fixed items it would just be a matter of using some coordinate system in the simulation. Addresses can also be used.

For non-fixed geo-referencing is dynamic. Their location in relation to other automata, represented in simulated goals, destinations, opportunities, etc., may be dynamic in space and time. It is also worth noting that there are instances in which geo-referencing is dynamic for fixed geographic automata also, for example when land parcel objects are subdivided during simulation.

The movement rule set, M, is also important.
Repel/Attract rules.
Obstacle negotiation
Collision avoidance
Flocking

Geographic Automata Neighbors and Neighborhood Rules

The set of neighbors of different types is necessary for the application of transition rules and movement, which depend on properties of geographic automata and their neighbors.

In contrast to the static and symmetrical neighborhoods in traditional CA models (figure 2.1a), spatial relationships between geographic automata vary in space and time and thus, rules should be formulated in such a way as to account for geographic automata locations neighbors at each time step in a model’s evolution.

GAS as an extension of GIS

GIS is a natural environment for preparing and visualizing GAS models. GAS are based on the ability of GIS to register data spatially, and to use spatial analysis to shape data as layers of entity-level objects and to estimate relationships between them.

GAS as an Extension of the Vector Model

Geographic automata of many types correspond to GIS features, which can be used to derive automata location. For fixed geographic automata, it is directly, by using the coordinate representation of a corresponding GIS feature.

The majority of relationships between geographic automata can be naturally evaluated within Vector GIS: Standard overlay operators such as point in polygon buffering, intersection, etc., make it possible to determine how automata are situated in relation to other automata. Also adjacency, contiguity, continuity, distance, accessibility, visibility, and so on.

GIS is excellent for visualizing and querying the outcomes of GAS simulations.

GAS and Raster Models
GAS is functionally connected to Raster in several ways: Each pixel of a raster can be regarded at least morphologically, as an automata cell, georeferenced by column and row positions.

The choice of a raster or vector view is beyond the GAS scheme and evidently depends on the goal of a model.

GAS as a Tool for Modeling Complex Adaptive Systems.

Urban systems are seen as complex adaptive systems.

GAS models of urban systems are built as collectives of interacting geographic automata and these interactions may be specified in such a way that they facilitate the emergence of higher-level entities, phenomena, and events, from the bottom-up.

-emergence
-bifurcations
-catastrophes

GAS models can be used to reflect phases on continuous quantitative development in urban systems as well as recognizing and modeling possible abrupt and qualitative changes in their dynamics.

The emphasis of GAS is on investigating self-organization in space.

From GAS to Software Environmental of Urban Modeling

Object Oriented Programming as a Computational Paradigm for GAS

Any Geosimulation modeling is ‘Simulation of GAS.’ Any Geosimulation should therefore provide functionality for the representation of all three GAS components:

1. To implement and Locate Urban Objects
2. To determines neighborhood and other spatial relationships between them
3. Formulate state transition, migration, and neighborhood rules.

The object oriented programming paradigm provides and excellent framework for facilitating these collectives of automata in a relatively seamless manner.

From an Object Based Paradigm for Geosimulation Software

Any simulation software environment must come equipped with some clear determination of its own position between the extremes of abstract concepts and rigid formalizations. Ideally, an environment for geosimulation should be ‘open’, as open as the concepts of urban dynamics it is used to simulate. However, the greater the freedom
afforded to the user, the lower the potential benefit to be gleaned from specialized simulation software.

Two fundamental features of a software environment for geosimulation should be as follows:

1. Openness for different formalizations of objects’ behavior, provided that the used accepts a GAS based view of the system;
2. Users’ full responsibility for the formalizations’ correctness.

Geosimulation is a modeling paradigm and its realization is bound to concept-oriented software. We argue that a reference to urban systems is sufficient to specify a universal studies in formalizing and studying dynamics of urban phenomena through simulation based experimentation and exploration.

GAS as Simulation Environments as Temporally Enabled OODBMS

A central problem in any Geosimulation software environment is the absence of a general solution for managing relationships between objects/entities. Partial solutions which depend on the nature of the objects and the semantics of their relationships have been developed to deal with this problem. The following are examples of such solutions.

Temporal Dimension in GAS

There are basically two ways to deal with time in a GAS. One is to have an external clock which causes transition rules to kick in based on the various interactions which might be happening between the entities in the geosimulation (synchronous mode of updating). The other is to have each automaton have its own little internal clock allowing for units of time to have different meaning for different automatons (asynchronous mode of updating).

Object Based Environment For Urban Simulations (OBEUS) – a Minimal Implementation of GAS

Three categories of classes in the OBEUS scheme:
1. Universal – Those considered that are necessary for simulating any urban process.
2. Model – Inherits abstract Universal Classes and is necessary for specifying any model of a specific class.
3. User Defined – Classes Reflect specificity of users’ models and are constructed anew, if necessary, in each application.

Management of Time

OBEUS architecture uses both synchronous and asynchronous modes of updating time.
In Synchronous mode all objects change simultaneously and conflicts arise when agents compete over limited resources, as in the case of two householders trying to occupy the same apartment. Resolution of such conflicts depends on the context of the simulation and the modeler.

In asynchronous mode, objects change in turn, with each observing the urban reality as left by the previous object. Conflicts between objects are thereby resolved; thus the order of updating is critical in determining the results.

Management of relationships in OBEUS

Leader and follower relationships are defined. E.g. landlord and renter. Leaders are responsible for managing the relationship.

Implementing System Theory demands

Self organization is often too important in urban systems to be ignored.
Geodomains are the simplest approach to emergence, determined by the set of a priori given predicates defined on geographic automata is implemented; domains are thus limited capturing ‘foreseeable’ self organization of specific types. E.g. rich or poor neighborhoods, industrial or commercial areas.

Expensive real estate making nearby area more valuable – industrial area decreasing residential property value, etc.

Verifying GAS Models

In comparing a simulation to the real world it is necessary to establish three things
1. initial conditions: location of fixed objects, the initial states of the entities in the simulations, etc.
2. Boundary conditions or spatial constraints: the conditions that need to be held during the entire simulation run.
3. The values of model parameters, which should be employed to run the simulation ahead of time the parameters that specify the rules of state transitions, migrations, and neighborhoods.
   - Pixel x Pixel comparison of maps
   - “intuitive tuning”
   - one can rerun the model with all possible values of parameters and maximize the correspondence to reality.
   - Dynamic Programming

Universality of GAS Models

Sufficient for the formalization of and analysis of any urban system. The logical chain between geographic systems and GAS representation is as follows:
Geographic system -> Priority of location information and spatial relations
Between elements -> collective dynamics of geographic automata in space.

The minimal GAS skeleton allows for a degree of standardization between automata models and other systems, not least of which are GIS. It also provides a mechanism for transferability. Until now, the majority of –if not all-urban simulation could be investigated only be their developers.

The development of a software environment for GAS would breach this barrier, potentially offering opportunities to turn urban modeling from art in to science.

2 additional steps for full implementation of GAS Framework:
1. should be transformed into a software environment
2. Simulation language based on GAS approach should be developed. The goal is to enable the formulation of simulation rules in terms of objects’ spatial behavior.