System Dynamics for Modeling Metabolism Mechanisms for Urban Planning

Conference Paper · June 2018

3 authors:

Reinhard Koenig
Bauhaus Universität Weimar

Martin Bielik
Bauhaus Universität Weimar

Sven Schneider
Bauhaus Universität Weimar

System Dynamics for Modeling Metabolism Mechanisms for Urban Planning

Reinhard Koenig[1, 2], Martin Bielik[3], Sven Schneider[3]

[1] Bauhaus-University Weimar
Professorship for Computational Architecture,
Weimar, Germany
reinhard.koenig@uni-weimar.de

[2] AIT Austrian Institute of Technology,
Smart and Resilient Cities,
Vienna, Austria
reinhard.koenig@ait.ac.at

[3] Bauhaus-University Weimar
Chair of Computer Science in Architecture, Weimar, Germany
martin.bielik@uni-weimar.de,
sven.schneider@uni-weimar.de

ABSTRACT

Dynamic urban development simulation models are usually separate to urban planning tools making it difficult to test the consequences of urban planning variants directly without switching between expert tools. This paper presents an approach to integrating system dynamics simulation at various scales and abstractions in the visual programming environment Grasshopper for Rhino3D. We demonstrate how Grasshopper may be used with additional customized components as a flexible integrated urban planning and simulation framework. For this purpose, we present three urban planning model examples: The first is a classical system dynamics simulation that abstracts from spatial elements. The second adds spatial relations in terms of distances between locations in a grid. The third shows how to represent a city in more details and adds a network analysis module for more precise distance calculations. As result, we demonstrate a highly flexible approach for integrating simulations for various aspects that predict the behavior of an urban system in order to facilitate more sustainable urban planning processes. The main drawback of this new level of flexibility is the relatively slow execution time for complex simulations.

Author Keywords

System dynamics; urban metabolism; urban planning; urban modeling; urban simulation; Grasshopper for Rhino

ACM Classification Keywords

I.6.1 SIMULATION AND MODELING; I.6.5: Model Development; I.6.7: Simulation Support Systems

1 INTRODUCTION

The motivation for this work was to contribute towards more sustainable urban planning practice by providing advanced digital planning assistance. We focus on the consumption of resources when building new cities or transforming existing ones. The explicit consideration of resource consumption is a relevant aspect for achieving sustainability development goal (SDG) 11: *Make cities inclusive, safe, resilient and sustainable* [22]. The fact that humankind consumes on average the resources of 1.7 worlds, and in highly industrialized countries even 3 up to 5 worlds (based on data from the Global Footprint Network National Footprint Accounts 2017) makes it clear how important this is. As shown by Rhode and colleagues [18], urban morphology and thus urban planning has a considerable impact on resource consumption, making it a critical factor for global sustainability.

We start from the concept of urban metabolism introduced by Richard Rogers [19], which proposes considering a city as a self-sustaining organism. To approach the challenge of estimating the resource consumption of urban planning proposals with digital models, we use system dynamics models as introduced by Jay Forrester, who represented cities [4] and finally the whole world as an abstract dynamic system. In contrast to Forrester's abstract models, we aim to model urban metabolism mechanism for very specific urban planning aspects. The stocks and flows approach as exemplified by Gerhard Schmitt [20], is a helpful analogy for representing relevant aspects of urban systems on the right scale in order to inform corresponding urban planning decisions on sustainability aspects. As a theoretical foundation, we understand urban metabolism as "*a model to facilitate the description and analysis of the flows of the materials and energy within cities, such as undertaken in a material flow analysis of a city. It provides researchers with a metaphorical framework to study the interactions of natural and human systems in specific regions"* [17].

The methodology for modelling urban systems that we introduce in this paper makes it possible to increase the level of detail and the spatial resolution as a planning project becomes more specific. We start with the description of an abstract urban system dynamics model that may be useful for planning the metabolic system for a region around a city including agriculture and energy production. For demonstrating the second level model, we introduce a grid-city in which citizens and urban functions are located in grid cells and make their location choice based on distances to other locations. The third level model includes the full geometric details of a city and therefore makes it possible to use network measures and specific capacities for individual locations. These three levels allow the user to successively extend an initial model with more detail, while using the same simulation mechanisms.

2 STATE OF THE ART

The first prominent advocate of a city as dynamic system was Jay Forrester with his "Urban Dynamics" model [4]. It focuses on temporal changes but completely neglects the spatial aspects. Forrester's model assumes an inner-city system as a closed system without relationships to its surrounding region. Inside this model city, one can test various hypothesis on the dynamic relationships between employments, residential location choice and economic developments.

Forrester's model, however, contradicts the first law of geography by Waldo Tobler *"everything is related to everything else, but near things are more related than distant things"* [24]. Therefore, most of the current urban models [1, 2, 30] are based on the causal structure introduced by Lowry [15], which, put briefly, suggests that the development of land uses depends on existing land uses and the distances between them. Traffic volume is modeled as interchange rates between land uses.

From urban economics, there are models that capture basic urban agglomeration forces. The two main opposing forces are a centripetal and a centrifugal force [10, 13]. Centripetal forces describe centralizing forces, which express the advantages and needs of a dense population (agglomeration effects). Centrifugal forces describe the advantages of decentralized locations, which are primarily the ample availability of space and the absence of polluting emissions. Krugman also introduced the first core-periphery model with multiple regions, which he called the racetrack economy [11]. In this model, several cities are arranged in a circle with each city connected only to its direct neighbors. Solving the model numerically made it possible to show the effects of scaling [12] on agglomerations of multiple cities or regions.

The ability to understand a complex urban simulation model quickly becomes very complicated when a few control parameters are introduced. Ideally, a sensitivity analysis shows how the parameter-space (containing all possible combinations of control parameter settings) is connected to the solution-space (the solutions for all parameter combinations). Daniel Zünd [33] demonstrates an interesting analysis method in his thesis that uses hierarchical clustering to show how probable it is that an urban simulation model ends up in a certain category or type of result.

A number of dynamic urban models have been implemented in the context of underlying research projects, but they are not available as tools for urban planning practice [1, 3, 8]. To overcome this situation, some simulation frameworks like AnyLogic [34] or NetLogo [31] provide a simplified programing language for implementing various kinds of simulations, and also offer a huge collection of simulation models. Unfortunately, they are not easily combinable with tools used in urban planning practice, where it is important to be able to change planning variants flexibly and to work with precise spatial representations. GIS based simulation frameworks like UrbanSim [28], or Urbanica [27] address this latter requirement but reviews of these land use transportation models compare their features [29] and criticized the concept of integrated urban simulation models [14, 23]. Nevertheless, urban planners rarely use them, because they lack the flexibility required for creative urban planning and design. This may be one reason for the recent development of more urban planning oriented software such as UrbanCanvas [35] or Urban Footprint [36].

For our simulation approach, we use the parametric, visual programming tool Grasshopper for the CAD system Rhino3D developed by David Rutten over the past few years. In the context of urban simulation, it is worth mentioning that parametric design tools have a very different history to GIS. Originally derived from the CAD realm, they permit the specification of geometric constraints within a specified parameter value range. Parametric design tools accept variable input data, establish mathematical relationships and produce further data, including geometric information. The combination of a parametric tool with the visual programming environment Grasshopper makes it possible to define very flexible urban models including geo-referenced and semantic information in a custom data model.

It is important to clarify that integration in our context means the possibility to combine various simulations at various scales and abstractions for the changing needs of an individual urban planning project. By contrast, the aforementioned integrated land use transportation models try to include all relevant aspects of an urban system in one comprehensive large scale urban model.

To analyze the street network in one of our examples, we use graph-based measurement methods to compute distance matrixes for all locations in the model. In addition to metric distances we use angular distances, which are based on the Space Syntax [5–7] theory and measuring method [25, 26]. The distances in the network graph are used to define the accessibility of a location, which can be more or less important for different urban functions.

3 URBAN DYNAMICS MODELS

The objective of developing the following models was not to come up with a very practice-oriented simulation for answering a specific urban planning question, but to offer a framework that can be easily adapted to various planning problems. Therefore, the following examples use abstract planning scenarios to illustrate the basic mechanisms of the simulations.

3.1 Wolf-Sheep-Predation Test Model

The first challenge was to translate the system dynamics modelling approach to the visual programming logic of Grasshopper. This is not a straightforward task, because the data flow model of Grasshopper is not designed for implementing loop-structures by default. To enable loops in Grasshopper, we used the plugin Anemone by Mateusz Zwierzycki. To test our implementation, we reproduced the wolf-sheep-predation system dynamics model from the NetLogo model library [32] in Grasshopper. We compared both model outcomes (Figure 1), to verify that the population-graphs display the same dynamic progression. The top row of Figure 1 shows the two different model representations in NetLogo and Grasshopper. The Grasshopper model differs to that of classical system dynamics diagrams, but if we consider just the main system dynamics part (in the green box of Figure 1), we can see there are only four main components used in the core model.

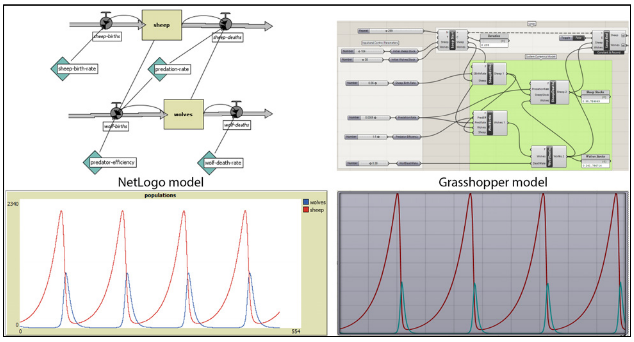

Figure 1: The population-graph from NetLogo (bottom left) shows the same dynamics as that of our Grasshopper implementation (bottom right). The top row shows the corresponding models implemented in NetLogo (left) and Grasshopper (right).

3.2 Abstract urban dynamics model

Based on the wolf-sheep-predation model we implemented an abstract urban dynamics model that includes the development of population size, number of jobs and food production (stocks). These stocks depend on various relationships between each other (flows) and on the available areas for the corresponding land use. The model structure is illustrated in Figure 3 as a classical system dynamics diagram and the translation of this model as a Grasshopper program is shown in Figure 2, with the core model inside the green box in the center of the definition.

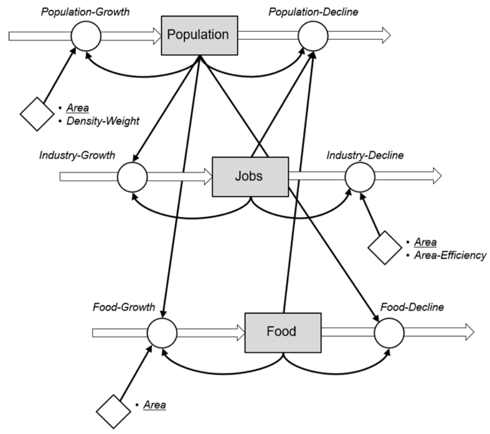

Figure 3: System dynamics diagram for the abstract urban dynamics model with the three stocks "Population", "Jobs", and "Food" and the corresponding flows between them.

Beside the control parameters of the model, the core of a system dynamics model are the flows that manipulate the stocks. In our implementation, the flows are modelled by the six Grasshopper components shown in the green box in Figure 2. For simple flow mechanisms, we use expression components and for more complicated ones the C# script component. An example of a simple expression for population growth is:

$$P_{(t)} = P_{(t-1)} + \left(P_{(t-1)} \;\; \omega \left(1 - \frac{P_{(t-1)}}{A_P}\right)\right) \tag{1}$$

where P is the population stock at time step t, ω is the density weight, and A_P is the available area for the population.

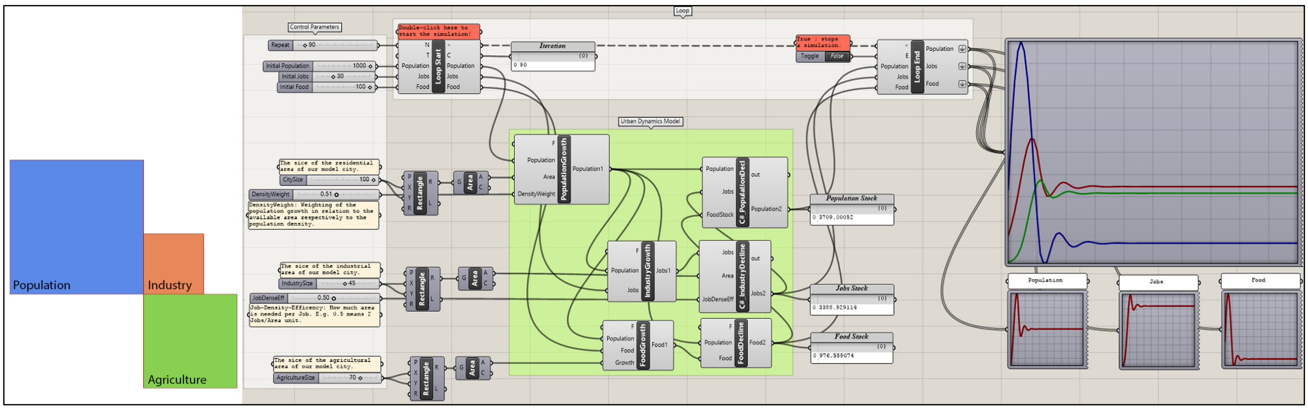

Figure 2: Abstract urban dynamics model implemented in Grasshopper. The core model components are shown in the green box in the center . The control parameters are listed in the left half. The colored boxes on the left visualize the area input values that restrict the growth or the stocks. The diagrams on the right show the dynamic behavior of the model (the development of the stocks).

The flow mechanism for population decline is more complicated, since it depends on the existing population, the available food and jobs:

$$\Delta J = 1 - \left(\left(-1 / \frac{J_{(t)}}{P_{(t-1)} + 1} \right)^2 + 1 \right)$$
$$\Delta F = 1 - \left(\left(-1 / \frac{F_{(t)}}{P_{(t-1)} + 1} \right)^2 + 1 \right) \tag{2}$$

$$P_{(t+1)} = P_{(t)} - \left(P_{(t)} \ \Delta J \ \ \Delta F \right) \tag{3}$$

The decline of population P in the next time (t+1) depends on the relationship of population stock P, the food factor ΔF, and the jobs factor ΔJ. The food factor ΔF expresses in essence the relationship between available food F for the population P and the jobs factor ΔJ expresses the relationship between available jobs for the population P. If there is not enough food or not enough jobs for the population, the population will decline.

We do not explain all flow mechanisms of the model because it serves only as an example implementation to demonstrate how easily and quickly the model can be adapted and extended by adding or adapting Grasshopper components.

3.3 Grid-city model

The next model includes some explicitly spatial aspects in its dynamics, which are based on the causal structure of the Lowry approach [15]. This means that the development of land uses depends on existing land uses and the distances between them. In our grid-city model, we use only two land uses, population and workplaces.

The presented model uses a 12 × 8 = 96 cells grid as shown in Figure 5. We begin by computing once the all-pair Manhattan distances between the cells. To process these distances, we use a data tree in Grasshopper as illustrated in the circle diagram in Figure 4, where each node of a branch contains 96 distance measures from a cell to all cells.

Based on the distances, we compute for each iteration the attractivity of a cell for new population or workplaces. The left-hand columns in Figure 5 shows two exemplary attractivity maps.

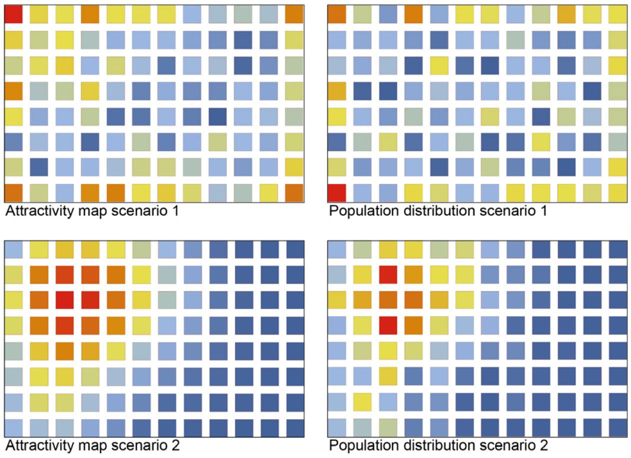

Figure 5: Grid-city model after t=100 steps. The diagrams on the left show the attractivity values for population, and on the right for population distribution. In scenario 1 on the top row, we assume that people prefer to be at a distance to one another, and in scenario 2 on the bottom row that people prefer to be located close to others. Blue represents minimal values and red maximum values.

To compute the attractivity of a cell, we first normalize all distances to the range [0; 1]

$$\hat{d}_{i,j} = \frac{d_{i,j}}{\max d} \tag{4}$$

where $\hat{d}_{ij}$ is the normalized distance between two cells i and j. The influence V of a land use on cell j to the land use on cell i is expressed by an exponential decline function, where near things have a greater impact than more distant ones. The

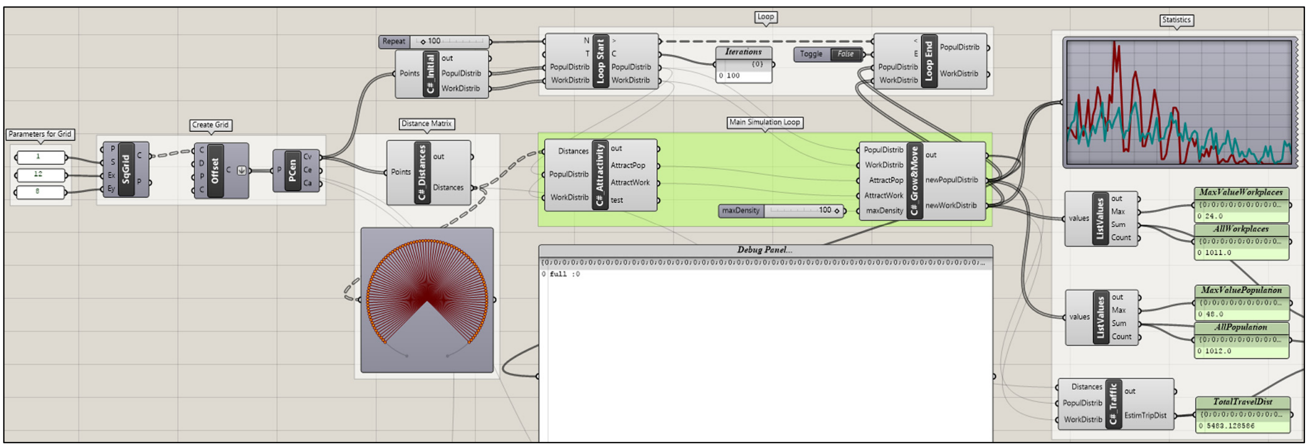

Figure 4: The grid-city model implemented in Grasshopper. In the green box in the center are the core model components. On the left side, the initialization of the grid and the distance computation is shown. The diagrams on the right side show the number of people (red) and workplaces (green) per cell.

following is a basic example of how population perceives other residential land use:

$$V_{i,j}^{PP} = \hat{d}_{i,j}^{\;-\alpha} \tag{5}$$

By the power factor α, we determine how fast distant uses become less important. In our example in Figure 4, we use $\alpha = 7$. The attractivity A for population P in a cell i is now the sum of all influence values V multiplied by the amount of people or workplaces that are located on a considered cell. For scenario 2 in Figure 5 we assumed that people like to live near to other people in order to gain advantages from central urban function:

$$A_i^P = \sum \left(V_{i,j}^{PP}\ P_j\right) \tag{6}$$

This attractivity function covers the perception population to population only. We can extend it easily to also include the population's need to be located near to or further away from workplaces. For the workplaces, in turn we use a corresponding attractivity function.

The last step of an iteration of the simulation is to add workplaces or people to the model. This means they need to select a certain location based on the attractivity map. For this purpose, we use the attractivity values A as weights for a roulette wheel selection:

$$\rho_i = \frac{A_i}{\sum_i A_i} \tag{7}$$

The weights ρ_i correspond to the size of a roulette wheel slot. Now, we generate a random number in the interval [0; 1] and test the slot it falls into in the roulette wheel. This slot defines the location that is chosen. Using the roulette wheel method, we distribute 10 people and 10 workplaces per iteration.

Figure 5, bottom row shows the result for scenario 2 after 100 iterations. In scenario 2, people attract people and workplaces want to be located close to the population. In each iteration, we add 10 people and 10 workplaces. The capacity of a cell is limited to 100 units, which can be any combination of workplaces and people. Following this model logic, we find one large population cluster and one corresponding attractivity cluster. Workplaces are not shown, but have a similar distribution as the population as visible in the density diagram in Figure 4.

Scenario 1 in the top row of Figure 5 shows the results for a model where people prefer to not be too closely located to one another. The attractivity function simply changes to:

$$A_i^P = \sum \left(\left(1 - V_{i,j}^{PP}\right)\ P_j\right) \tag{8}$$

while the rest of the model remains unchanged. The population distribution in Figure 5 has no center and the corners of the map, which are farthest away from everything, tend to be the most attractive locations for the people.

In order to compare the two scenarios presented in Figure 5 we introduce a simple traffic estimation function:

$$T_i = d_{i,j}\ P_i\ \frac{W_j}{\sum_j W_j} \tag{9}$$

The traffic T generated by population P in cell i is computed as product of the amount of people P in this cell, the distance $d_{i,j}$ to a workplace W_j and the weight of workplace density at the distant cell j, which is calculated as number of workplaces at a cell divided by all workplaces in the model. To characterize a scenario, we simply add all traffic values T_i to a T_{sum} value. Using this traffic estimation, we obtain the following values for the two simulation scenarios in Figure 5:

Scenario 1: $T_{sum} = 6468$

Scenario 2: $T_{sum} = 5051$

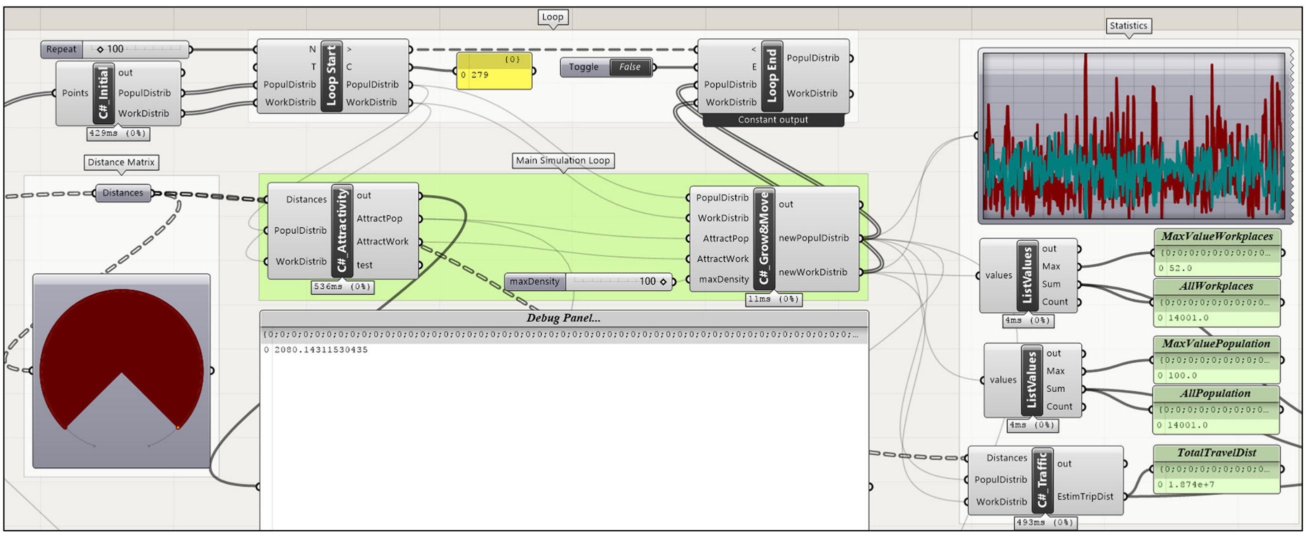

Figure 6: The net-city model implemented in Grasshopper. The core model components are in the green box in the center. On the left is the initialization of the population and workplace placements and a reference to the network distances. The right diagrams show the number of people (red) and workplaces (green) per cell after 100 iterations.

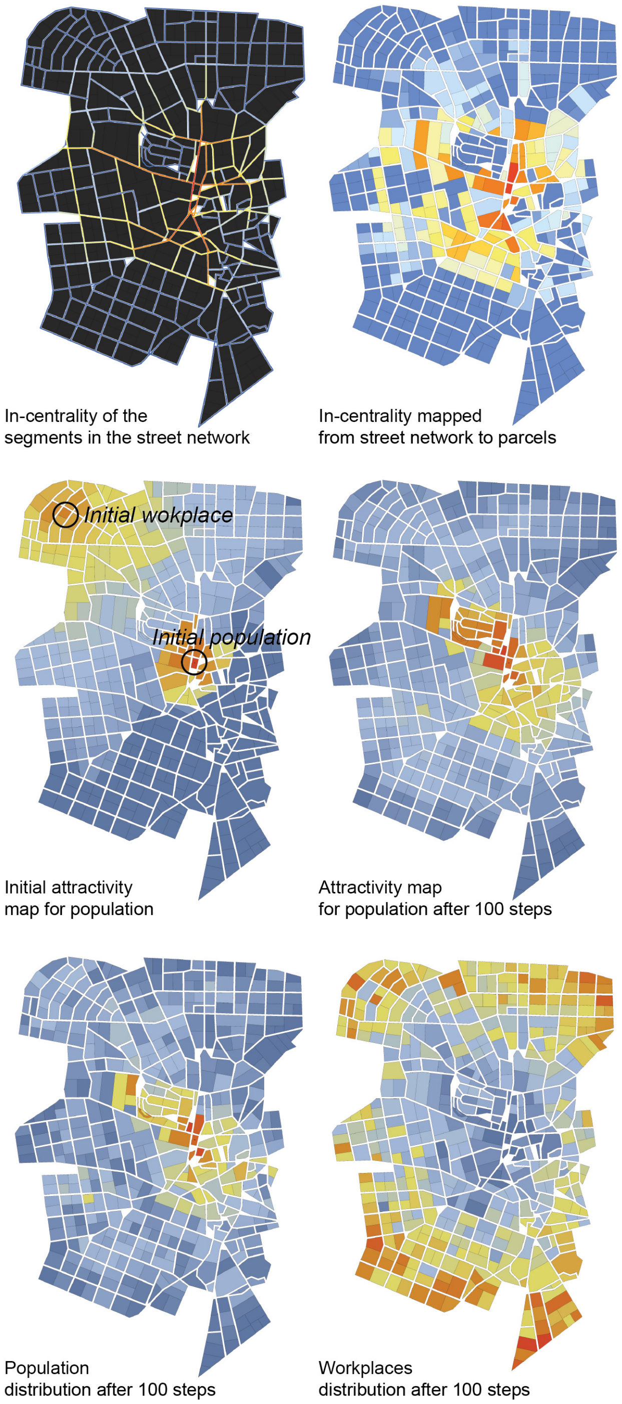

Figure 7: Net-city using Weimar as the base map. Top row: In-centrality measures computed for the street network (left), and centrality values mapped to the parcels (right). Middle row: Attractivity maps for the initialization of the simulation (left) and after 100 steps (right). Bottom row: Distribution of the population (left) and the workplaces (right). For all maps, blue represents minimal values and red maximum values.

This means that distributions with separate clusters of workplaces and population produce less traffic than distributions where workplaces and population are more randomly mixed. The random mix in scenario 1 results in 1417 more trips, corresponding to approximately 27% more traffic than in scenario 2. We did, of course, assume a random link between the workplace location and place of residence of a typical citizen. Nevertheless, this result would merit some more detailed investigations on the effects of mixed-use cities in contrast to the modernist principle of separation of land uses.

3.4 Net-city model

We called this model net-city because, in contrast to the grid-city, we use now the real geometry of a city and compute the distances between locations by means of shortest paths in the street network. For this demonstration, we used the central part of the city of Weimar in Germany. The Grasshopper model for the simulation is very similar to that of the grid-city (compare Figure 4 and Figure 6).

The network centrality measures are computed using the DeCodingSpaces toolbox for Grasshopper [37]. It makes it possible to combine metric (meters between two locations) and angular distances (angular turns between two locations) [26] to compute the shortest paths in a network. We use a weighting of 0.8 for angular and 0.2 for metric distances. The in-centrality measure of a location is the sum of this location with respect to all other locations (in our case street segments) in a network. The resulting centrality map is shown in Figure 7, top row. The underlying all-pair shortest paths in the network replace the Manhattan-distances from the grid-city model in our simulation (Figure 6).

Based on the in-centrality map, we compute the attractivity of each location. Therefore, we adapt equation (6) to add nearness to workplace as an attractivity factor for the population:

$$A_i^P = \sum \left(V_{i,j}^{PP}\ P_j + \gamma\ V_{i,j}^{PW}\ W_j\right) \tag{10}$$

The factor V^{PW} expresses the perception of workplaces W by population P. It is computed in the same way as V^{PP} in equation (5). The weighting factor γ relates both influence vectors V to each other. For the example model run in Figure 7 we used $\gamma = 0.1$. The resulting population distribution in Figure 7 shows that the people prefer to be located in the center, because the distances to other people as well as to workplaces are in average shortest here. This is the nature of high in-centrality values in a network and the reason why the population distribution tends to follow the centrality map.

For the attractivity map of workplaces, we use the equation:

$$A_i^W = \sum \left(\left(1 - V_{i,j}^{WP}\right)\ P_j + \beta\ V_{i,j}^{WW}\ W_j\right) \tag{11}$$

which means, that workplaces are ideally set apart from the population to minimize exposure to the emissions they may produce. At the same time, workplaces are attracted by each other to share infrastructure. To define their attractivity map,

A^W workplaces must therefore consider the population P, expressed by V^{WP} as well as the workplaces W, expressed by V^{WW} and the weighting factor β, for which we chose $\beta = 0.5$. The V^{WW} force has a relatively small impact on the resulting distribution of workplaces but leads to some additional clustering (Figure 7).

4 DISCUSSION

The simulations introduced here are certainly far from being realistic. The models we present can rarely reproduce a realistic distribution of population and workplaces, since we abstract from too many other factors. However, the model is very good for revealing certain potentials for developments assuming various scenarios. We take the view that every good model needs to abstract from many real world aspects in order to reveal some relevant basic mechanisms [21].

However, the fact that the model dynamics and the underlying rules are very abstract makes it difficult to find a meaningful relationship to some real urban development mechanisms or to reflect certain planning policies. Simulation models always require some translation from reality to the abstract model world. The meaningfulness of the result always depends on the purpose of the model and the question that needs to be answered.

Usually, some randomness determines the result of simulation models such as the one we present in this paper. While the results are similar in every model run, there are differences in detail. This can be approximated by running the model many times to determine an average or most probably outcome. At the same time, as discussed above, we should be aware that the simulation results represent only an abstract tendency of the model and it is therefore not meaningful to investigate the model behavior in too much detail.

5 CONCLUSIONS AND OUTLOOK

We mentioned at the outset that the modelling approach introduced here has the advantage of being able to start with a much reduced, abstract, non-spatial model and to successively add more and more details. Comparing the three models we present here, we can see that the first one exhibited a more complex dynamic behavior, represented in the graphs of changing stocks. In contrast, the two last models have just a linear growth of stocks (workplaces and population), because there is no mechanism implemented that would reduce the stocks. This aspect could be added, by limiting the attractivity using some density relationships or by introducing a threshold of attractivity, below which workplaces or people would leave the city. This, however, goes beyond the scope of introducing a basic dynamic modeling approach for Grasshopper that makes it possible to use urban development simulations more interactively within urban planning and design processes.

In this paper, we have tried to prove that using a very simple model with only two land uses can already produce some meaningful results in terms of indicating more or less attractive locations for the analyzed land usages. We can use the model now to test urban planning options that immediately show the corresponding effects on the potentials of locations. In a future extension, we may use the potential of parcels as input value for the next level of detailed urban design in order to specify the building typology. For this purpose, we plan to apply procedural building models [9, 16].

The simulation methodology presented here is of most potential for the planning of new cities, for example in fast-developing countries in Africa, South-East Asia, or South-America. It intends to overcome the current situation in which urban planners develop sustainable metabolistic concepts for complex urban systems without the tools to check their functionality quickly and easily.

ACKNOWLEDGMENTS

The idea for combining urban metabolism with advanced urban simulation methods originates from Gerhard Schmitt and the original concept for the Future Cities Lab (FCL) at the Singapore ETH Centre (SEC). This concept was included in a design studio by my colleagues at the Bauhaus-University Weimar and gave rise to the need for the integrate simulation framework for urban design assistance that we have discussed here. I owe a great deal of thanks to the students and teachers in this studio and their projects and discussions which served as a basis for the work presented in this paper.

All models presented in this paper can be downloaded at: *http://decodingspaces-toolbox.org/urban-simulation-with-grasshopper/*

REFERENCES

1. Batty, M. 2005. *Cities and Complexity: Understanding Cities with Cellular Automata, Agent-Based Models, and Fractals*. MIT Press.
2. Batty, M. 2013. *The New Science of Cities*. The MIT Press.
3. Dynamische Simulation und Visualisierung räumlicher Prozesse: *http://www.entwurfsforschung.de/alt/RaumProzesse/DSVPro.htm*. Accessed: 2017-12-30.
4. Forrester, J.W. 1969. *Urban Dynamics*. MIT Press.
5. Hillier, B. 2007. *Space is the machine: A configurational theory of architecture*. Space Syntax.
6. Hillier, B. 2007. The Common Language of Space: A way of looking at the social, economic and environmental functioning of cities on a common basis.
7. Hillier, B., Leaman, A., Stansall, P. and Bedford, M. 1976. Space syntax. *Environment and Planning B: Planning and Design*. 3, 2 (1976), 147–185.
8. Koenig, R. and Bauriedel, C. 2009. Generating settlement structures: a method for urban planning and analysis supported by cellular automata. *Environment and Planning B Planning and Design*. 36, 4 (2009), 602 – 624. DOI:https://doi.org/10.1068/b34025.

9. Koenig, R., Miao, Y., Knecht, K., Buš, P. and Mei-chih, C. 2017. Interactive Urban Synthesis Computational: Computational Methods for Fast Prototyping of Urban Design Proposals. *Computer-Aided Architectural Design. Future Trajectories*. G. Çağdaş, M. Özkar, L.F. Gül, and E. Gürer, eds. Springer. 23–41.
10. Koenig, R. and Mueller, D. 2011. Cellular-automata-based simulation of the settlement development in Vienna. *Cellular Automata: Simplicity behind Complexity*. A. Salcido, ed. INTECH. 23–46.
11. Krugman, P. 1993. On the number and location of cities. *European Economic Review*. 37, 2–3 (Apr. 1993), 293–298. DOI:https://doi.org/10.1016/0014-2921(93)90017-5.
12. Krugmann, P. 1991. Increasing Returns and Economic Geography. *Journal of Political Economy*. 99, (1991).
13. Krugmann, P. 1996. *The Self-Organizing Economy*. Blackwell.
14. Lee, D.B. 1973. A requiem for large scale modeling. *Journal Of The American Institute Of Planners*. 39, 3 (1973), 163–178.
15. Lowry, I.S. 1964. *A Model of Metropolis*. Rand Corporation.
16. Miao, Y., Koenig, R., Bus, P., Chang, M., Chirkin, A. and Treyer, L. 2017. Empowering urban design prototyping: a case study in Cape Town with interactive computational synthesis methods. *Protocols, Flows and Glitches, Proceedings of the 22nd International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA)* (Hong Kong, 2017), 407–416.
17. Pincetl, S., Bunje, P. and Holmes, T. 2012. An expanded urban metabolism method: Toward a systems approach for assessing urban energy processes and causes. *Landscape and Urban Planning*. 107, (2012), 193–202. DOI:https://doi.org/10.1016/j.landurbplan.2012.06.006.
18. Rode, P. et al. 2017. *Resource Urbanisms: Asia's divergent city models of Kuwait, Abu Dhabi, Singapore and Hong Kong*.
19. Rogers, R. 1997. *Cities for a small planet*. Faber and Faber.
20. Schmitt, G. 2015. *Information Cities*. ETH Zurich, Chair of Information Architecture.
21. Starfield, A.M., Smith, K.A. (Karl A. and Bleloch, A.L. (Andrew L.. 1994. *How to model it : problem solving for the computer age*. Burgess International Group.
22. Sustainable Development Knowledge Platform: Sustainable Development Goal 11: 2017. *https://sustainabledevelopment.un.org/sdg11*. Accessed: 2017-12-29.
23. Timmermans, H. 2006. The Saga of Integrated Land Use-Transport Modeling: How Many More Dreams Before We Wake Up? *Moving Through Nets: The Physical and Social Dimensions of Travel*. K.W. Axhausen, ed. Elsevier. 219–248.
24. Tobler, W.R. 1970. A Computer Movie Simulating Urban Growth in the Detroit Region. *Economic Geography*. 46, (1970), 234–240.
25. Turner, A. 2001. Angular Analysis. *3rd International Space Syntax Symposium* (Atlanta, 2001).
26. Turner, A. 2007. From axial to road-centre lines: a new representation for space syntax and a new model of route choice for transport network analysis. *Environment and Planning B: Planning and Design*. 34, 3 (2007), 539 – 555.
27. UD_InfraSim Urban development simulations & infrastructure demand assessment - AIT Austrian Institute Of Technology: 2017. *https://www.ait.ac.at/en/research-fields/smart-and-resilient-cities/projects/ud-infrasim/*. Accessed: 2017-12-30.
28. Waddell, P. 2002. UrbanSim: Modeling Urban Development for Land Use, Transportation, and Environmental Planning. *Journal of the American Planning Association*. 68, 3 (Sep. 2002), 297–314. DOI:https://doi.org/10.1080/01944360208976274.
29. Wegener, M. 2004. Overview of land use transport models. *Handbook of transport geography and spatial systems*. 5, (2004).
30. Wegener, M. 2005. Urban Land-Use Transportation Models. *GIS, Spatial Analysis and Modeling*. D. Maguire, M. Batty, and M.F. Goodchild, eds. ESRI Press. 203–220.
31. Wilensky, U. 1999. *NetLogo*. Center for Connected Learning and Computer-Based Modeling, Northwestern University.
32. Wilensky, U. 2005. *NetLogo Wolf Sheep Predation (System Dynamics) model*. Center for Connected Learning and Computer-Based Modeling, Northwestern University.
33. Zünd, D. 2016. *A Meso-Scale Framework to Support Urban Planning*. ETH Zurich.
34. https://www.anylogic.com/
35. http://www.urbansim.com/platform
36. https://urbanfootprint.com/
37. http://decodingspaces-toolbox.org/