The Report Committee for Sadra Dehghan Hosseinabadi
Certifies that this is the approved version of the following Report:

Utilization of Computational Planning and Geo-Design as a Tool for having Interactive Conversations in Scenario Planning

APPROVED BY
SUPERVISING COMMITTEE:

Robert Paterson, Supervisor

Robert Stepnoski

Utilization of Computational Planning and Geo-Design as a Tool for having Interactive Conversations in Scenario Planning

by

Sadra Dehghan Hosseinabadi

Report

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science in Community and Regional Planning

The University of Texas at Austin
May 2018

Acknowledgements

My gratitude goes to my readers Robert Paterson and Robert Stepnoski for their mentorship and guidance through the process of this project. I would also like to thank Niloufar Karimipour for being an amazing friend and keeping me motivated through the duration of this project. A final thank you to the amazing Austin office Perkins+Will Cities and Sites team for inspiring the main idea for the project and educating me through gaining experience by working on great real-world projects, such as the site chosen for the hypothetical place of the scenarios built, through my time interning with them.

Abstract

Utilization of Computational Planning and Geo-Design as a Tool for having Interactive Conversations in Scenario Planning

Sadra Dehghan Hosseinabadi, M.S.C.R.P.
The University of Texas at Austin, 2018

Supervisor: Robert Paterson

In the process of planning and designing in an urban context, building and calculating information to report for the substantial number of elements in the urban environment is a very time consuming and labor-intensive process. Advancement of computational or parametric design tools has enabled designers and planners to script in automated tools for the repetitive procedures that should have been done manually in the past. Although more time consuming at the beginning, this approach accelerates modeling numerous elements in a certain project greatly, as it provides the opportunity to create multiple varying iterations and alternatives for a certain project in a matter of seconds.

Attempting to automate the recurring and time-consuming process of building out masses and reporting back data, an automated scenario builder is created which utilizes procedural modeling to do those tasks. This could help create an automated scenario builder that can be utilized as a tool for engaging conversations with clients and stakeholders during visioning exercises. Multiple development typologies are coded within the scenario builder, so the tool could create alternative scenarios in real time based on

feedback from the clients. The automated scenario builder can have a reporting function built in to evaluate various scenarios on the go to see how each alternative meets the vision that has been intended for the future of a place.

The main advantage of this approach is that it allows for creation of an ever-growing comprehensive library of toolsets which could be adapted for each project consultants encounter based on the context and individualities they have. Growth of the library leads to having a better toolset which with the advancement of technology will be able to take on a larger number of tasks within the planning process. However, it is crucial to note that automating the process of building out scenarios is not a computerization of the design and it is just a tool for planners or designers to streamline the process.

Table of Contents

List of Figures

Chapter 1: Introduction

Understanding how a physical plan will impact an area is an integral part of the planning process. Geospatial technologies are getting more and more integrated into the process of local development planning process and how people vision and visualize the future of a place. GIS technologies have very strong 2d platforms which are used extensively in planning practices, however, 3d tools are very much underutilized in the profession.

Recently, there has been some shifts moving towards a more regular use of 3d modeling in land development planning to visualize proposals and demonstrate the spatial impacts of it. CityEngine, a Geospatial 3d modeling software, which utilizes a procedural modeling process for creating masses of the urban environment, is one of the tools used in order to create 3d models of cities quickly, using GIS data and coded scripts as rules to create the mass based on the data associated with a piece of land.

Creating a generative planning model that can adapt to multiple conditions of various sites is a strong benefit of utilizing CityEngine for creating 3d cities. It is a strength that has been noticed by the movie and gaming industries for modeling large scale urban environments for the use of computer-generated imagery (CGI) technology within their work. Being able to model existing conditions or proposals designed with hand crafted data sets is a strong capability that the software presents to urban planners/designers, but can the toolset within the software be utilized to create an automated scenario builder for development proposals?

Figure 1: In Independence Day: Resurgence, aliens destroy a 3D replica of Singapore built using CityEngine. (Copyright 2016 20th Century Fox)

Having the ability to generate scenarios automatically and rapidly, could be a very strong tool for consulting firms in order to be able to have a more engaging conversation with the clients or stakeholders of a specific project. Designers are usually constrained by time limits for the number of variations of a scenario they could present to a client, and a large part of the conversation of how various versions would be, happens between the designers and consultants without ever being presented to the clients. Having an automated scenario builder in their toolkit would enable designers to showcase more alternatives to the stakeholders and also show their edits to them with a real-time platform that would implement some minor/major edits instantly.

Computational planning and procedural modeling

Being called by many names such as parametric design, generative modeling, computational design, or procedural modeling, an algorithmic approach to 3d modeling has enabled designers and graphics to work faster and make more complex patterns. Many software are available for utilizing this technique varying mostly based on the scale of work and the end product needed. Grasshopper, a plug-in for Rhino, and Dynamo for use in architectural design and CityEngine for urban design/planning practices are some of the most commonly used software in today's practices.

Modeling and visualizing large urban areas is a costly challenge for computer graphics, needing thousands of hours of technician labor (Muller & Parish, 2001). Utilizing an algorithmic approach towards building out the urban fabric is one of the techniques used to create models of the built environment in a faster and less costly manner. This approach can be more time consuming at first, scripting the parameters shaping the built environment, but as the scale and number of iterations of the model increase, the more beneficial the upfront overhead will be (Esri R&D Center, 2017).

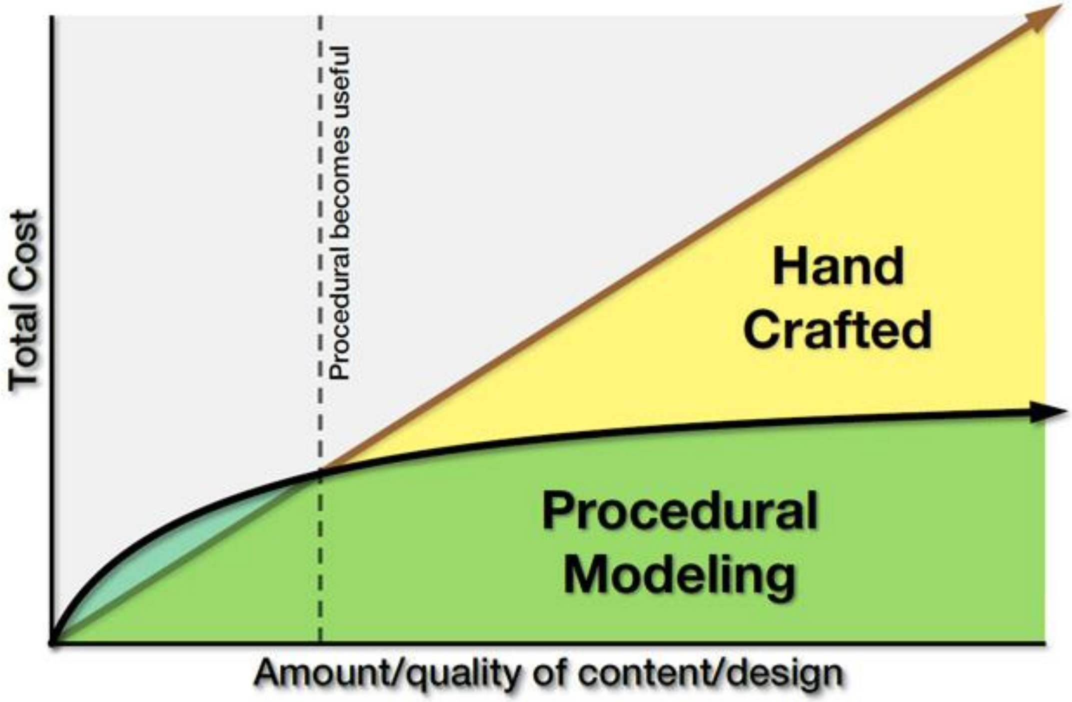


Figure 2: Total cost of modeling methods, Source: (Esri R&D Center, 2017)

Although an algorithmic approach to the process of 3d modeling a city seems to be streamlining the planning process with computer scripting, creating an automated scenario builder poses a set of challenges within the built environment. Physical attributes of the urban fabric are created by transportation networks that follow the population, environmental features influencing the shape of the place, history of the area, culture, etc. (Muller & Parish, 2001). Some of these shaping parameters can be coded into a script with ease, but some features could be undefinable for computers with today's technology where the context of the human life and behavior is considered.

In this report, the capabilities and limitations of an automated scenario generating script, coded in CityEngine, is tested to see how land use zonings can be coded for urban areas in order to model the physical form of the future of proposals. In an attempt to create/computationally design a scenario for an urban development, many parameters are scripted into a code, with a built in real-time reporting feature, to simulate the factors shaping the built environment.

Methods

This report consists of 6 major parts which following the introduction, will be a literature review on scenario building in general and factors considered in the planning process. In the third section, the basics of past attempts of algorithmic 3d modeling in urban design and the foundation of this approach to creating the built environment are reviewed. This chapter of the report will also look into the attributes of computational planning and shaping factors of the possible development that can be built out. This includes investigating what features need to be included within the code, what elements of the urban environment shape the physical build out, and tools that exist for scripting the automated scenario building code.

The fourth section of the report will be introducing a site and evaluating the physical context of the land. In order to eliminate political, cultural, and historical context, which can’t be coded in, a hypothetical location for the site is assumed, although it’s an actual real-world tract of land in Jefferson Parish, Louisiana. For the purpose of creating the automated scenario builder, a list of typologies are set as the defining factors of the possible proposal for scripting those within the software.

The 5th chapter will be focused on the development of the automated scenario builder which will be a 3d buildout tool, creating what a future development could be for the site based on the set of goals and restrictions which are set in the development scenario. Finally, a section of this report will be assigned to present a summary of findings and evaluating capabilities or limitations of an algorithmic approach to urban design, based on modules and typologies. Looking into the benefits this approach could have for practicing urban planners and designers, future applications for consulting efforts taking this approach are considered and future research applications are described.

Chapter 2: Scenario building

In order to create an automated scenario builder, the first step is reviewing the basics of scenario planning and what scenarios are. Hopkins and Zapata describe scenarios as stories envisioning how the world changes and will be changing at some future time. Scenarios identify the issues and forces shaping communities resulted from a conversation with the actors within a community. Throughout comparing scenarios, one will be identified as the preferred one, however, other scenarios should not be ignored because development of scenarios are not completely within our control (Hopkins & Zapata, 2007).

Scenario planning is a strong tool for envisioning the future without having to predict it. Individually a scenario will be demonstrating a possible future, however, multiple scenarios in a set will be reflecting the multiplicity, complexity, and unpredictability of factors shaping the future of the urban settings (Hopkins & Zapata, 2007).

Scenarios are not predictions, projections, plans, nor are they forecasts. They are closest to simulations, providing a communicable set of structured, probable future systems. The term scenario has been used in many situations with varying meanings, and with some overlap between what is actual definition of a scenario is. Some authors have used scenario and vision in place of one another (Costanza, 1999), however, scenarios are not what a vision is in nature. Forces beyond the control of an organization shape a scenario, whereas visions are formed by what an organization wants to become or make happen. Indeed, if "vision typically answers the question, whom do we want to be or what we want to do, then scenario answers the question, where will we have to do this"? That defines kinds of environments which people will be trying to become whom they want to be as well as doing what they want to do (Hopkins & Zapata, 2007, cited as (Schwartz, 1991)).

The main goal of scenario planning is to tease out the important ways that the coming future could be different from what has been in the past. The process of building a scenario focuses on grounded explanations of change, creativity, and interacting with constituents as a source of knowledge and legitimacy. Success within the process of scenario planning stems from attracting the most publicly visible local leaders and implicit advocates (Hopkins & Zapata, 2007).

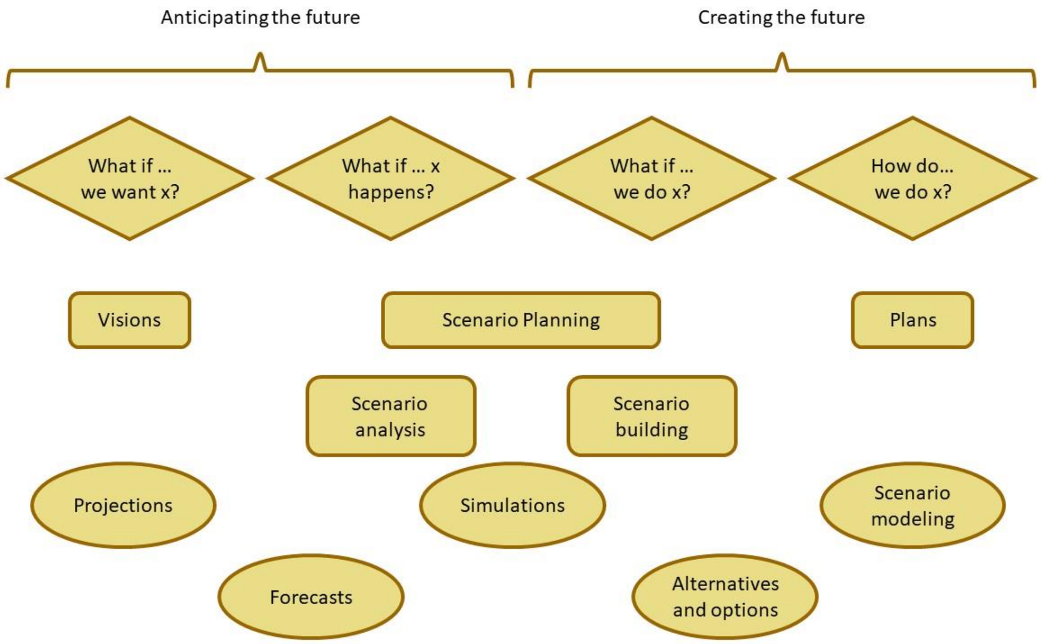


Figure 3: Rough relations among planning terms, Source: (Hopkins & Zapata, 2007, cited as (Gobal Business Network, 2004)*)*

Using scenarios to make urban plans

There are many reasons to do scenario planning. A main one is to focus the attention of an organization on the external world, to consider ways the world might change that are beyond their control. Second reason is that looking out into the future is very confusing with numerous factors impacting the world over the coming years, most of which have several uncertainties associated with them. Illuminating and examining the future is another reason for engaging in scenario planning. Perceptions of what the future will be, is usually largely based on what the past has brought and in an ever-shifting world, for reaching a certain vision, the strategies taken to reach the vision may change (Hopkins & Zapata, 2007, cited as (Bruchell, Downs, McCann, & Mukhreji, 2005)).

Scenario planning rests on four major principles: taking the long view, thinking from outside in, including multiple perspectives, and telling stories. Taking the long view basically means considering long-term periods of time such as five, ten, or even longer periods of time in order to allow significant systemic changes to take place, although changes usually happen more quickly than expected. Considering the forces and factors beyond our control is the main reason for thinking from the outside, because a single entities visions and decisions are not always what shapes the future. Building up on the multiplicative returns of various new perspectives interacting with each other, resulting in very creative and imaginative ideas, is the main point of including multiple perspectives. Finally, in order to be more communicative with the lessons, results, and ongoing learning from a set of scenarios, planners approach scenario planning with telling stories of what the future could be (Hopkins & Zapata, 2007 (Bruchell, Downs, McCann, & Mukhreji, 2005)).

Communication of scenarios are a very important part of scenario planning as the audience who has not been involved with the process of creating the scenario set, need to be able to envision and visualize them. Audience interaction with scenarios can be a very powerful experience and having it as an integral part of the process, would greatly streamline reaching the goal of scenario building of changing minds instead of making plans. It can be argued that scenario building is an experiential learning process and their strength will be waivered if they're simply delivered as to having an interactive development process with the stakeholders (Hopkins & Zapata, 2007).

A set of believable, relevant, challenging, and divergent stories is an imperative part of scenario building. This is necessary for having a tool which can be utilized for decision making, prioritization, and testing various alternatives while having a proactive conversation with stakeholders. Proper use of scenarios can complement other planning processes while adding perspective and insight to them, as well as monitoring for deeper shifts in the external environment and acting appropriately (Hopkins & Zapata, 2007).

Scenarios have a primary purpose of enabling safe and productive conversations among stakeholders on difficult issues, which they're very well suited for as they're hypothetical outlines and not actual plans. Ideally, these conversations will result in hope, cooperation, and are proactive while creating more unity and diminishing pessimism through simple, open acknowledgment of differences and by laying out story lines to see how far things might go in the future (Hopkins & Zapata, 2007, cited as (Heijden, 1996)).

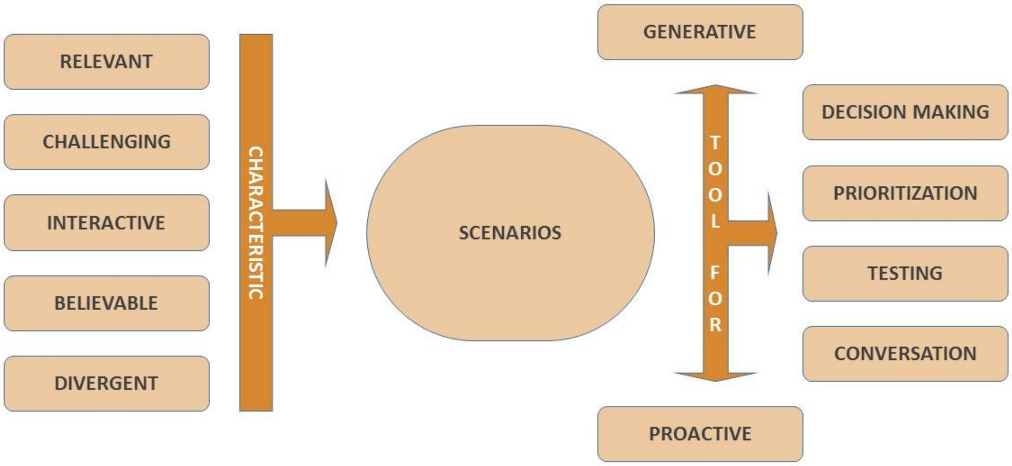


Figure 4: Characteristics of scenarios, Source: (Hopkins & Zapata, 2007)

Matching a probable future with a desired one is an initial step for building planning scenarios. Vision developments is very commonplace in planning practice and people carry visions of what the future will be based on past experiences. It is impractical to solely focus on plausible futures without the consideration of desired futures. In order to accomplish this, two parallel processes of analytical with sets of limits on the range of possible futures, and analysis of stakeholder values and goals run simultaneously, with the processes overlapping in some parts. For example, facts, trends, constraints, and issues are presented to stakeholders, as well as being elicited from them, because this will affect their awareness of the anticipated future. In other words, "this is an informed kind of visioning exercise" (Hopkins & Zapata, 2007).

This approach allows for identifying the problems and analyzing the context before meeting with stakeholders for setting evaluation criteria, but goals and objectives are up for discussion instead of being derived previously. The driving factor for the conversation is the conflict of goals and objectives which are combined in various ways shaping numerous scenarios in contrast of each other. This allows for an amalgamation stakeholder values and an analysis of driving forces shaping the goals of the future vision (Hopkins & Zapata, 2007, cited as (Heijden, 1996)).

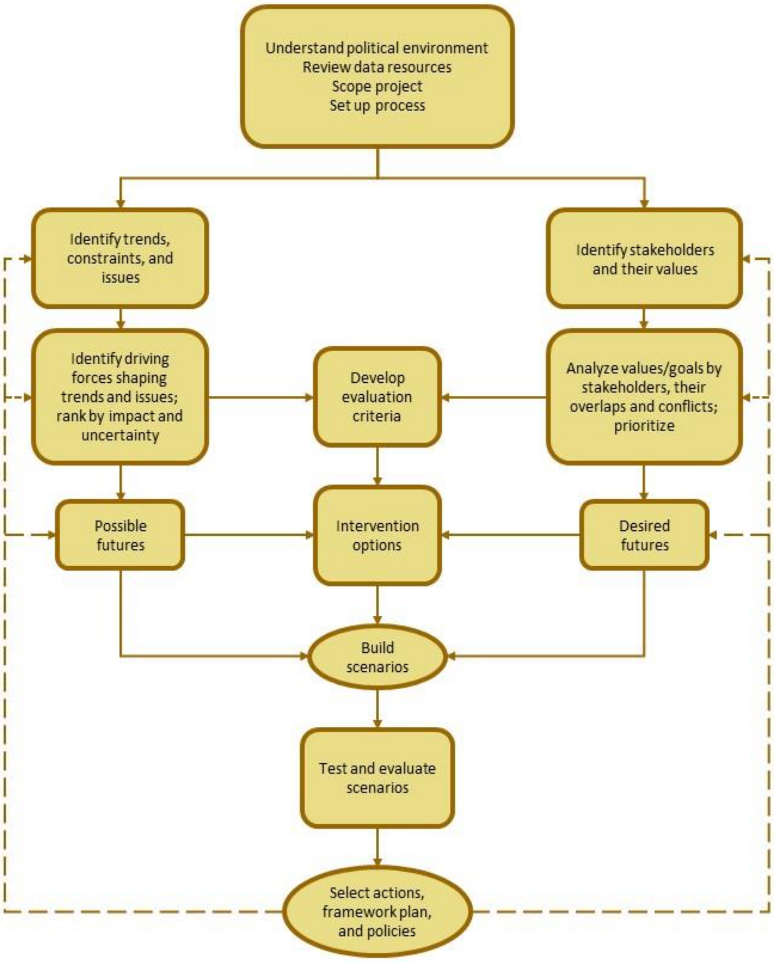


Figure 5: The scenario-building process, Source: (Hopkins & Zapata, 2007, cited as (Avin & Dembner, 2001)*)*

Tools for visualizing scenarios

Matching the tools utilized with the needs in the planning process for visual stimulation, is the first and the most important step in visualizing scenarios (Kwartler & Longo, 2008). After functionality and capacity for each tool should be established in order to build the capacity of interaction between various tools, which is referred to as interoperability of software (Kwartler & Longo, 2008). Since various tools function well for some specific functions at different scales, it is important to be able to cover abroad range of criteria by utilizing multiple tools. “The best tools are those that can help make a complex problem or set of issues sensible by providing the means to explain complexity rather than over simplification” (Kwartler & Longo, 2008).

> *The ideal tool for visualization of scenarios should be able to address five main functions which are integrating quantitative and qualitative issues, supporting formulation of performance indicators, evaluating performance on the fly, promoting identification of design options, and simulating design choices visually (Kwartler & Longo, 2008).*

For visualizing existing conditions, GIS is probably one of the most powerful tools available to planners as it has a database of existing conditions imbedded into drawings. There are numerous 2d and 3d GIS platforms which can incorporate information and analyses about a place in numerous moments in time, which can be ideal for tracking changes and sequential criteria-based evaluations. GIS datasets are typically static, representing a moment in time, however, integration of data through multiple tables or matrices (Kwartler & Longo, 2008, cited as (Bosselman, 1998)). Having a good understanding of the limitations of GIS is crucial in interpreting information as well as having the understanding that homogenous data is not available for every place within a study’s boundary (Maantay & Ziegler, 2006).

Although working with limited sets of data may limit the sophistication of what can be done with analysis, it still enables the stakeholders to acquire useful information with thematic maps and 3d visualizations. Three-dimensional representations of information may be still images of 3d build outs or a dynamic visualization where the user could experience the movement through the space either through preset camera paths, such as video walkthroughs, or freely move through the virtual environment, like a videogame, or be immersed in the space and be able to move throughout the place with technologies such as Virtual Reality (VR) (Maantay & Ziegler, 2006). The best visualization experience is in a 3d form because poeple experience the world in three dimensions and adding the third dimension to visualizations can be very helpful. Sometimes visualizations in 2d could be misleading until the 3d version is presented (Kwartler & Longo, 2008).

Visualization needs to go beyond modeling existing conditions and make the scenario or the design accessible to stakeholders(Kwartler & Longo, 2008, cited as (Christopher, Ishikawa, & Silverstone, 1977)). Design of a place translates policies imbedded within the visioning process which is rooted in the world of everyday experience. In some situations, visualization is used to show not a certain existing place but what a place could be in the future. In order to test out policies and design of a place designers can simulate a representation of a place that stakeholders can not actually locate but resembles the place they are familiar with. This technique is intentionally utilized to visualize what the vision of the place could be translated in design format and the character of the place without specifically alarming stakeholders with negative impacts on their properties and interests (Kwartler & Longo, 2008, cited as (McLuhan, 1967)).

Providing equal access to information and data to all stakeholders is an ideal outcome of visualizing designs and scenarios. Having the visualization tools as an integral component of the public process providing transparency can better provide access to information in the process for the public from the consultants. Providing transparent and understandable information to the stakeholders enables them to make more informed and confident decisions which will elevate the planning process into a broader exercise of stewardship (Kwartler & Longo, 2008).

Considering the characteristics of scenarios and ideal tools for visualizing them it could be concluded that an automated scenario building tool should be:

- Interactive with the stakeholders and consultants
- Generative and Proactive
- Adaptive to various contexts
- Telling a story in multiple platforms of visualization for better feedback
- Quantitative and Evaluative for examination of scenarios
- Dynamic to evolve with feedback
- Complex and Simplified

Chapter 3: Procedural modeling

Understanding the basic grammar of the tools that are being utilized is an essential part of creating a successful automated scenario building tool for streamlining the planning conversation with stakeholders. CityEngine uses a procedural modeling process, which is an algorithmic based approach towards 3d modeling, for massing the build out of the elements within an urban environment. The computational method utilized with CityEngine can offer valuable support with rapid generation of variables and basic reporting of the outcomes of the masses built.

Having a high number of elements building out an urban environment is the initial challenge for adopting a computational process for building out cities. There are many relations between these elements influencing the shaping and performance of each other. In order to simplify recognition of elements and creating a process for coding in the building blocks of the urban fabric, the elements shaping the physical form of the urban environment have been categorized to roadways, land use, blocks, parcels, and footprints of building (Muller & Parish, 2001).

The process of shaping a certain place starts with indication of street centerlines within the boundaries of the place which leads to creation of blocks within the enclosed areas between the streets. Each street centerline carries certain characteristics imbedded within, including the width of Right of Way (ROW), width of sidewalks on each side of the street, number of lanes, setbacks for buildable areas, etc.

Each block created by the ROWs will need certain attributes shaping them into parcels which will be containing the physical masses of buildings. Blocks are factionalized into different uses and parcels by zoning boundaries and parcel lines which in an existing context are drawn from GIS data but in a hypothetical one, mathematical operations will split the block into parcels and human interaction with the code or a random number will assign the use to parcels.

With the formation of parcels the buildable are in each could be defined by assigning the setbacks with other limiting factors within the land use definition for each area. In hypothetical scenarios, for each type of use there’s a certain required footprint area that needs to form in the buildable area, which mathematical operations can look for the optimal rectangle, or other geometries defined in the script if needed, to place within the buildable area. Created footprints can be assigned multiple factors such height, number of floors, design factors such as placement of openings, roof type, etc. (Muller & Parish, 2001). The culmination of all these elements shapes the extent of the urban environment and numerous factors or attributes shaping the 3d context could all be coded in, within the interface built in to the CityEngine software.

Simply put, creation of a representative series of blocks in an urban environment is based on the elements creating the visual organization of the block. This includes blocks, street and ROW, sidewalks, lot sizes, building types, and landscape element that conform to various shapes and uses of land. In existing places, sampling the GIS data can create a series of blocks representing lot sizes, frontages, building footprints, and abstract or photorealistic buildings, building a 3d model that has the look and feel of the place (Maantay & Ziegler, 2006). However, for creating non-existential places, factors for shaping the blocks and the rest of the sequential elements have to be defined, which can be achieved by coding mathematical scripts following physical parameters set for the place or setting a step for consultant interaction with the automated scenario builder to input variables manually for better achieving the vision of the place (Talton, et al., 2011).

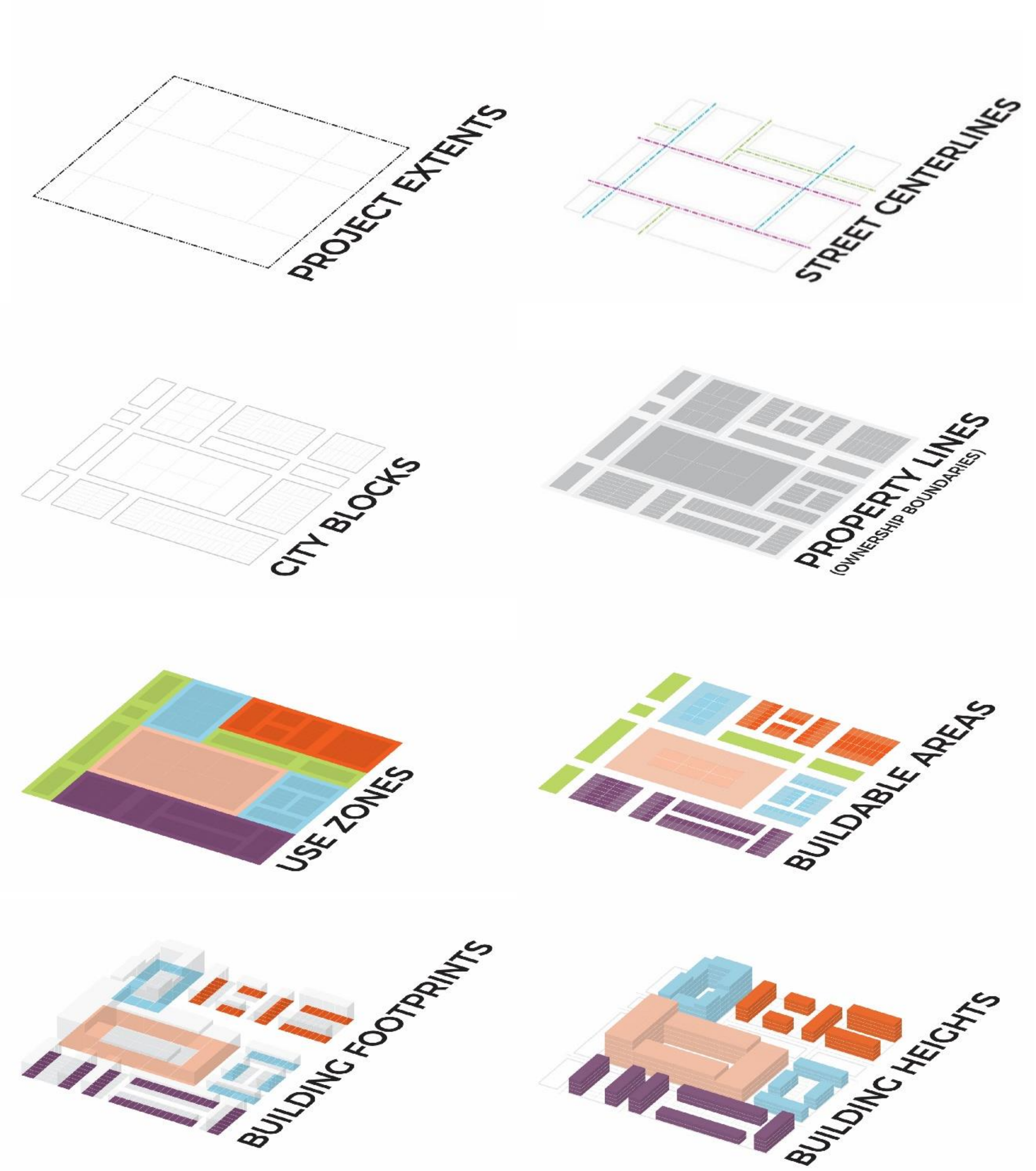


Figure 6: Elements of the urban fabric, Source: (Perkins+Will, 2018)

CityEngine and Generating Large-Scale Urban Layouts

For generating the layouts of large-scale urban environments, CityEngine employs the main elements of the built environment, with having street centerlines, blocks, parcels, and footprints as the main elements shaping the buildout of all elements. To model masses and elements more efficiently, CityEngine takes a procedural approach to 3d modeling instead of the classical method of intervention of the user, who manually interact with the model. In this method the computer is given a code-based procedure, as a shape grammar, which abstractly describes the interventions the user would employ in the classical method to model 3d geometries (Esri R&D Center, 2017).

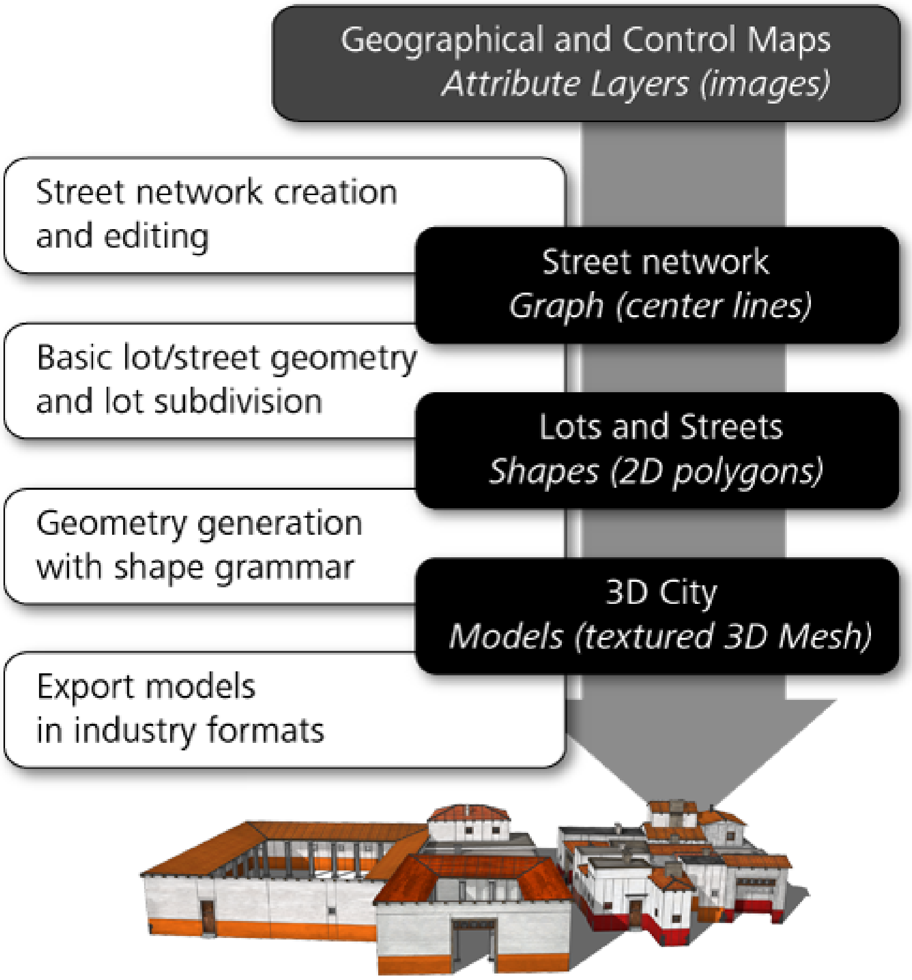


Figure 7: Overview of the CityEngine modeling workflow, Source: (Esri R&D Center, 2017)

Creating a dynamic layout of built environment, the user has a powerful tool to generate an interactive street network creating streets, sidewalks, and whole blocks which form the whole urban context in a dynamic format which gets updated in real-time as edits are made to the model (Esri R&D Center, 2017). The real-time modeling capability of CityEngine makes it an ideal tool for having a visioning/designing conversation with stakeholders and clients, while the data interoperability makes it a good match with most common formats utilized in planning, design, and visualization industries.

In addition, a reporting function can be coded into the mass generating script to be able to report back data regarding the development in real time and compare multiple scenarios. The information that can be coded includes any form of data correlated with the quantitative aspects of the built spaces such as gross area by use, additional units, population, employment, infrastructure details, etc.

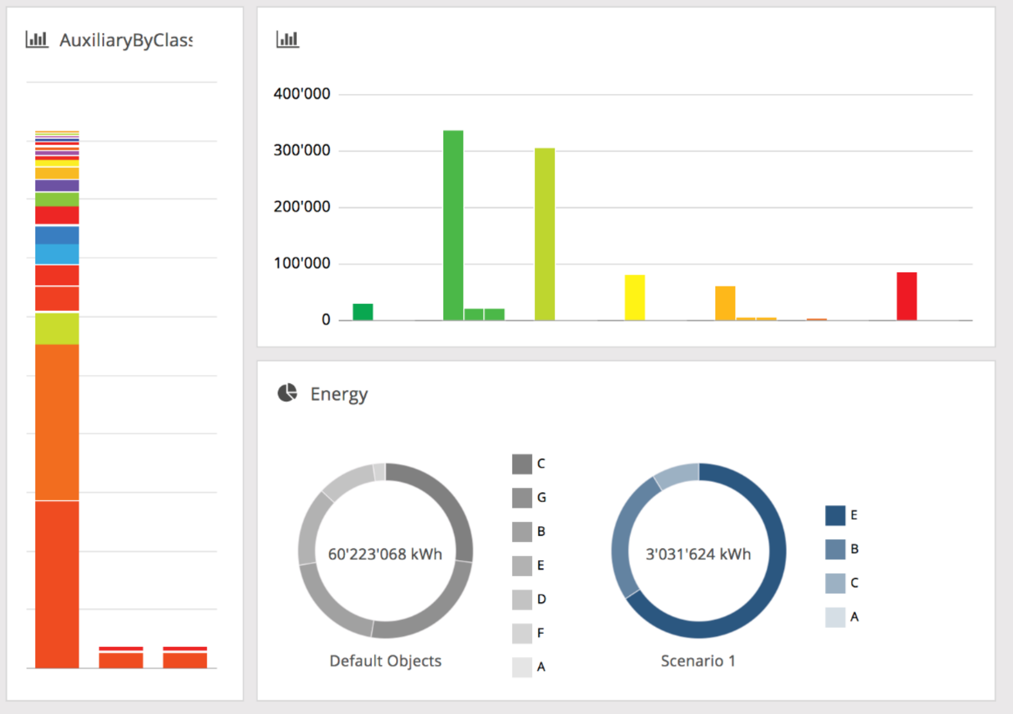


Figure 8: Reporting back in CityEngine, Source: (Esri R&D Center, 2017)

Computer Generated Architecture (CGA)

Having the elements of the urban environment set up, either through existing GIS data or creating them for non-existing scenarios, creating the 3d masses of the urban fabric is all that is left for acquiring a 3d representation of the place. CityEngine employs the CGA shape grammar which is a unique programming language designed to generate 3d architectural content with an iterative approach. This approach first builds out general massing for all the features in the first step, adds some details in the intermediate iterations, and finally applies all the levels of details defined in the code in the last phase of computing the program.

This iterative approach gives us the benefit of being able to simplify the process based on computational power of available devices, therefore it's possible to create simplified masses with everyday devices and if highly detailed or photorealistic visualizations are required devices with higher computational capacity could be employed. The workflow of generating masses and 3d models with CGA within CityEngine is as follows (Esri R&D Center, 2017):

1. Lots or parcels of land are imported from GIS data or created by the user within the software capabilities of block subdivision. Bounding masses of a building could also be the starting point for modeling buildings.

2. The user defines or assigns a CGA rule to create the building. In this part one rule could be applied to all the buildings or various rules could be assigned to different buildings.

3. In this step the user initiates the generation of buildings on selected shapes. In case of large models it is best practice to not generate every feature together due to memory constraints.

4. Editing the massed buildings can be done by, editing the rules, overwriting the parameters of the rules, using a different seed number in case of rules with random parameters used in them, or the local edit option provided in the 2017 version which allows users to edit a single feature in place.

5. At last after finalizing the design there are various options provided for exporting the models for brining into multiple visualization platforms such as gaming engines or directly export the basic texturing created in the software for a VR experience.

Figure 9: Generation of building geometries with CGA shape grammar, Source: (Esri R&D Center, 2017)

Creating an Automated Scenario Builder with CityEngine

Comparing capabilities of CityEngine with the requirements of a scenario building process many common grounds are found with being generative, adaptive, and dynamic. However, it is important to keep in mind that automated scenario builder is utilized as a tool for having a conversation with the stakeholders. Therefore, it is important to create opportunities for interaction of the user and manual manipulation of the built-out scenarios, so feedback from the clients could be implemented on the fly.

For the purpose of simplifying the conversation it is important to code in a simple massing function within the script, to make sure the elements discussed are clear to everyone. Having a basic massing, maybe with color coding of basic masses, will help better visualize uses and functions of envelopes of space, however as representative of the functioning of the space they are, they are not very helpful with visualization of the place.

Therefore, an extra function of building in basic urban design elements will be added on to the rule file for better visualization of the place, after the first steps of the discussion are cleared. Also, it is important to build in a reporting function in the script for the quantitative characteristics of the place just to be able to evaluate various scenarios on how the meet the visions and goals of a place.

Considering the requirements for a scenario building process and the needs for having a more productive conversation with the client, the automated scenario builder should have the following characteristics:

- Generative, adaptive, and dynamic
- Interactive to implement elements from the feedback
- Have manual editing capabilities built in
- Simplify and clarify the conversation
- Be able to produce basic visualization of the urban environment
- Have a reporting interface for clients to be able to see if their needs are met

Chapter 4: Visioning the building blocks

The first step for creating an automated scenario builder is visioning the basic element forming a certain type of development, which in case of this methodology utilizing CityEngine would be Right of Way (ROW) profiles and building types. The main elements of an ROW profile such as ROW width, lane width, sidewalk width, etc. could be coded in the Street CGA rule to model various streets within the development. For creating the various building types, it is required to script the siting and massing of each building type to be developed in each block formed by the ROWs. In this report various building uses suitable for an innovation district are examined which will be laboratory, office, residential, retail, and mixed-use buildings.

ROW profiles

The main focus of this report is on the site developments which will be modeling the masses of buildings, however, in order to create a successful scenario builder, having a street network within the site with hierarchy of ROWs is needed. For simplifying the creation of the street network, elements such as medians, bikeways, landscaping, etc. are eliminated and only travel lanes, curbs, and sidewalks in the street rule are coded, although all elements could be scripted in.

In this exercise the main streets will be six-lane streets with 11' travel lanes and 14' sidewalks on each side. The secondary streets will be four-lane streets with 11' travel lanes with 10' sidewalks, and the local streets would be two-lane streets with 12' travel lanes and 10' sidewalks on each side.

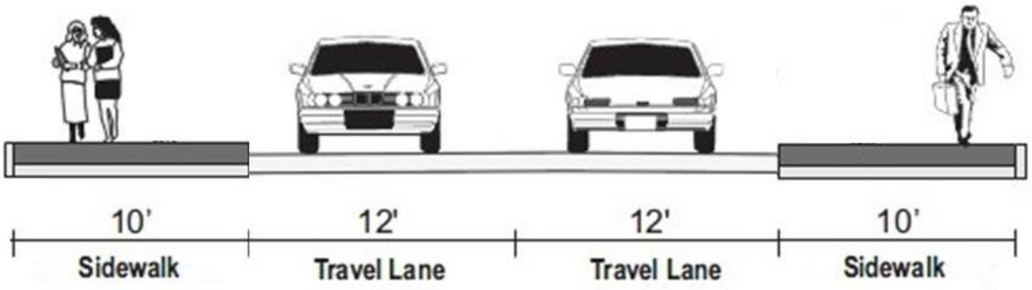


Figure 10: Two-lane street section within the area

Building types

As it is crucial for any successful place making plan to have a variety of uses and building types to be present in the development, a vital part of an effective automated scenario builder is having a diverse set of development typologies to choose from. For this exercise a and set guidelines for uses that would be suitable for a hypothetical innovation district development are chosen. These building typologies would be office, laboratory, residential, retail, and mixed-use buildings. Deriving building massing codes from the 2009 design guideline for Coldstream Research Campus for University of Kentucky, the forming factors for massing and height for each building type are set.

Office buildings

Two types of office buildings will be coded in for having the variety of choosing to have a medium-density option for normal office space and a high-density choice for more prominent office uses. The medium-density choice will be called commercial office and the high-density option will be named interstate office to represent the standout office buildings which will be developed in the innovation district.

The commercial office buildings having a maximum footprint of 20,000 sq. ft., will have a parallel façade to the main ROW sitting on a minimum 10-20 ft. setback to form a consistent building edge alignment. The building heights shall be up to 3 12 ft. stories with a predominant flat roof feature, allowing additional height for pitched roofs and requiring a minimum 40% transparency of the façade.

The interstate office building will have a maximum 35,000 sq. ft. footprint and a 10-20 ft. setback to have a consistent edge parallel to the street with other buildings on the block. The building will be between 2 and 8 stories, each 14 ft. high on an 18 ft. high pilot floor. A minimum 40% transparent façade will be required and having special features in the massing of the building for it to stand out within the district is recommended.

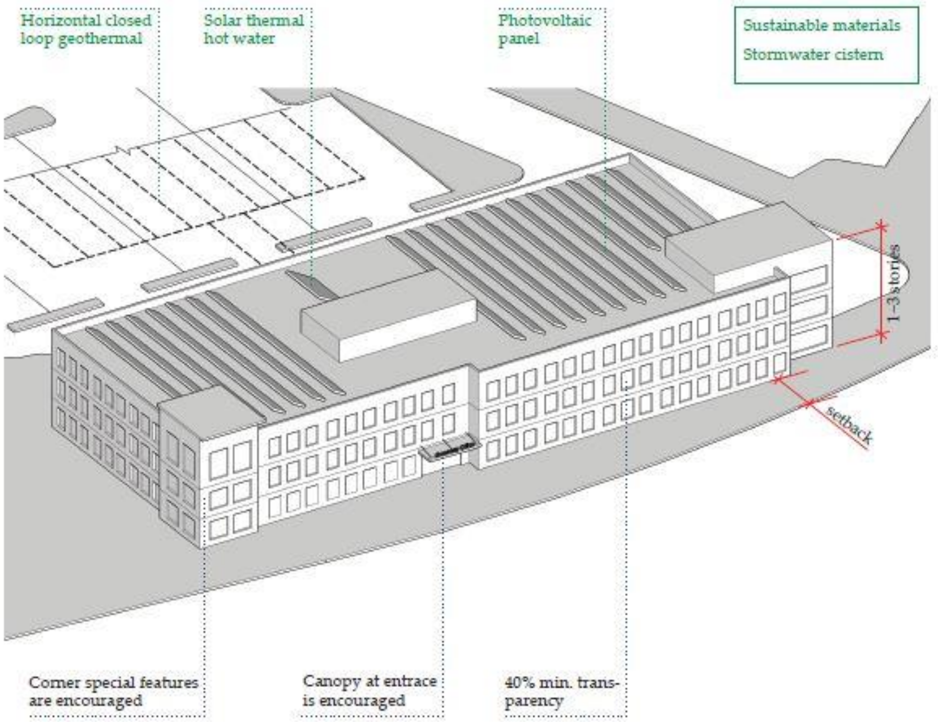


Figure 11: Commercial office buildout example, source: (EDAW Inc., 2009)

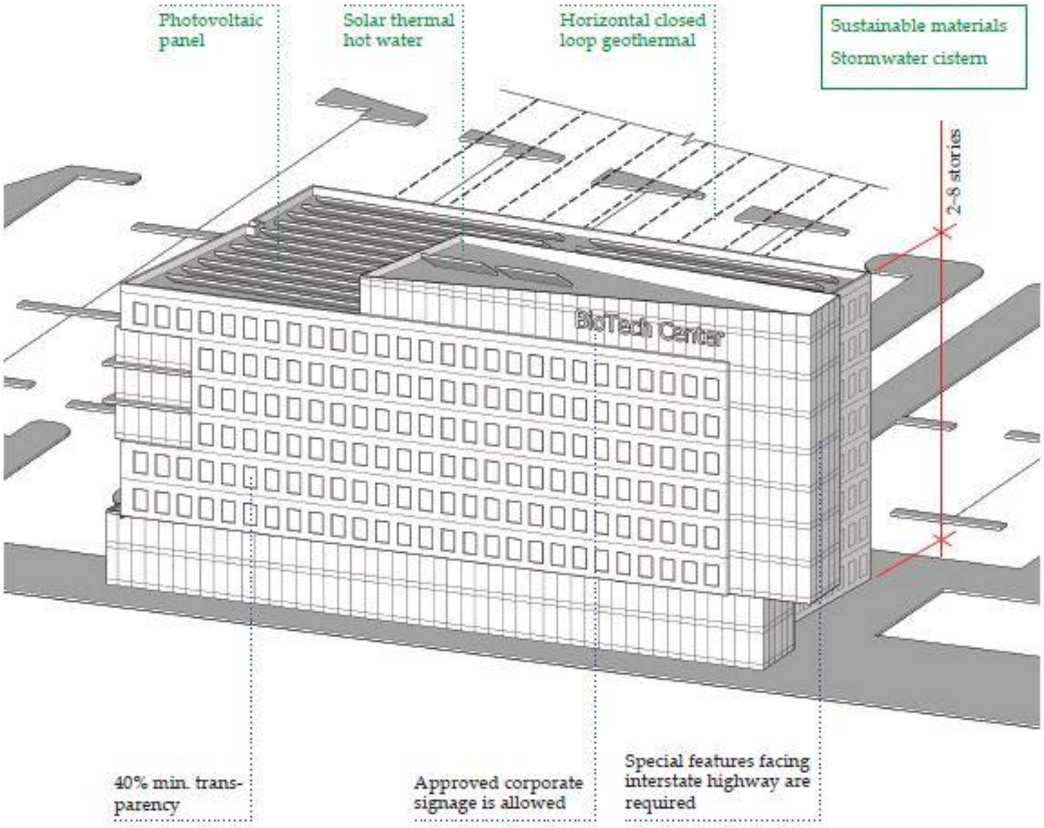


Figure 12: Interstate office buildout example, source: (EDAW Inc., 2009)

Laboratory buildings

As the tool being built is for having a planning conversation for an innovation district with the clients, it is important to build out a variety of office space and having laboratory buildings for cutting edge research. The laboratory buildings will settle in a maximum 30,000 sq. ft. and have a 10-20 ft. setback to align with other building frontages. The buildings should be up to 3 stories high with each floor being 20 ft. high and have a minimum 20% transparency on the facade.

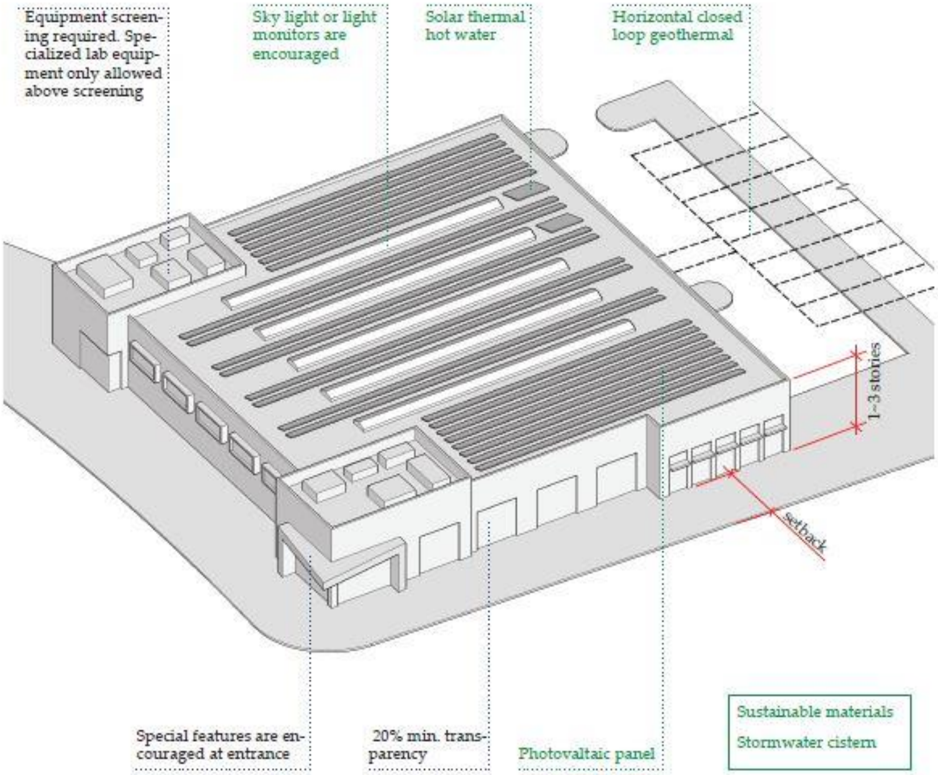


Figure 13: Laboratory buildout example, source: (EDAW Inc., 2009)

Residential buildings

Having a diverse supply of medium to high-density is important for the functionality of a successful innovation district. In this exercise creating an option of multi-family residential and a townhouse choice for single family housing is essential as every complete community needs a set of choices for housing. The multi-family residential buildings should rest on a maximum 12,000 sq. ft. footprint with a setback aligning with the block frontage. The buildings should be 3-5 stories high with a 10 ft. height for each floor. Townhouses will have a maximum 2,000 sq. ft. footprint and be 2 or 3 stories high with a minimum 25% transparency on the outer shell of the building. Townhouses could have a flat or gable roof depending on the building technology acquired for construction.

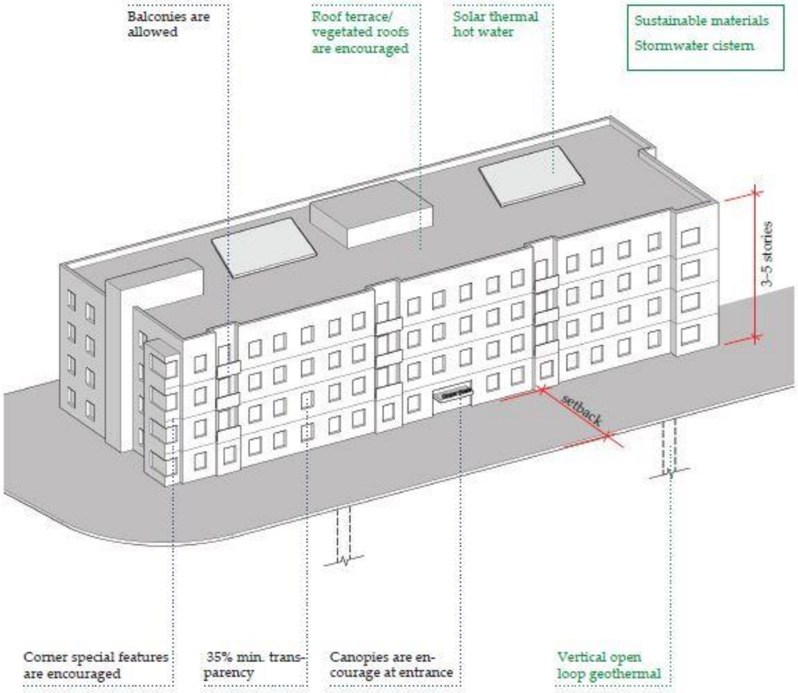


Figure 14: Multi-family residential buildout example, source: (EDAW Inc., 2009)

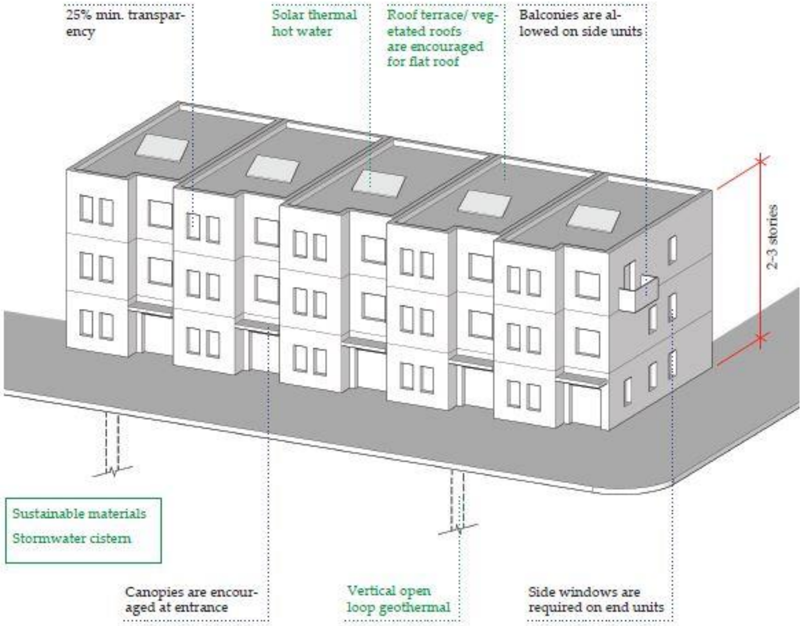


Figure 15: Townhouse buildout example, source: (EDAW Inc., 2009)

Retail buildings

Although it is better to have retail mixed in with other uses on the ground floor, with a horizontal mixture of uses, it is helpful to have a retail only option built into the code as to have the choice while having a conversation with the client and stakeholders. The retail buildings could sit on a wide variety of footprint sizes with a setback to align with the building frontages on the block. Retail buildings should be 22-35 ft. high but there won't be stories defined within the retail buildings to provide the fluidity for individual design. Roof of the retail buildings could be flat or gable depending on individual design preferences.

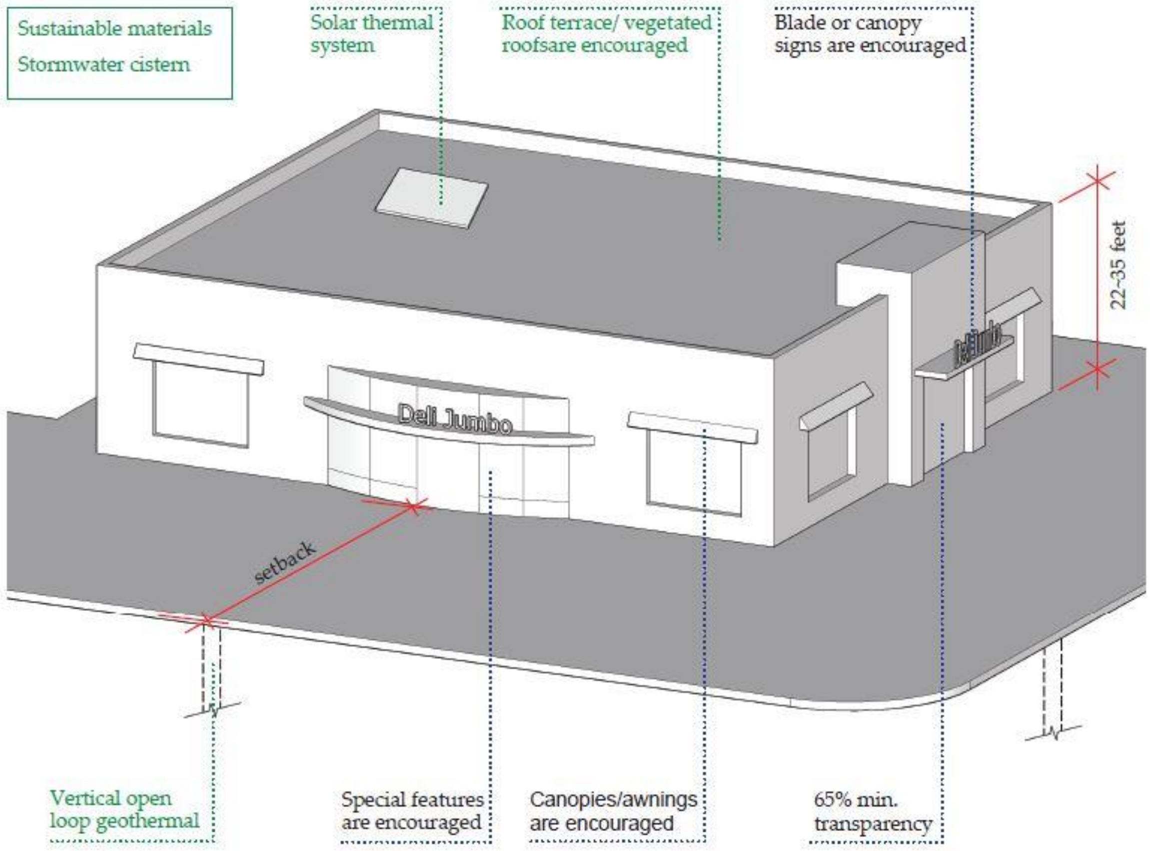


Figure 16: Retail buildout example, source: (EDAW Inc., 2009)

Mixed-use buildings

One good option for providing retail space within an active innovation district is having mixed-use buildings with pilot floors with commercial use embedded into the office and multifamily residential buildings. The ground floor pilot retail space should be 18 ft. high and have a minimum 65% transparency on the frontage. The additional 3-5 stories will be built out by the guidelines already set for commercial office and multi-family residential.

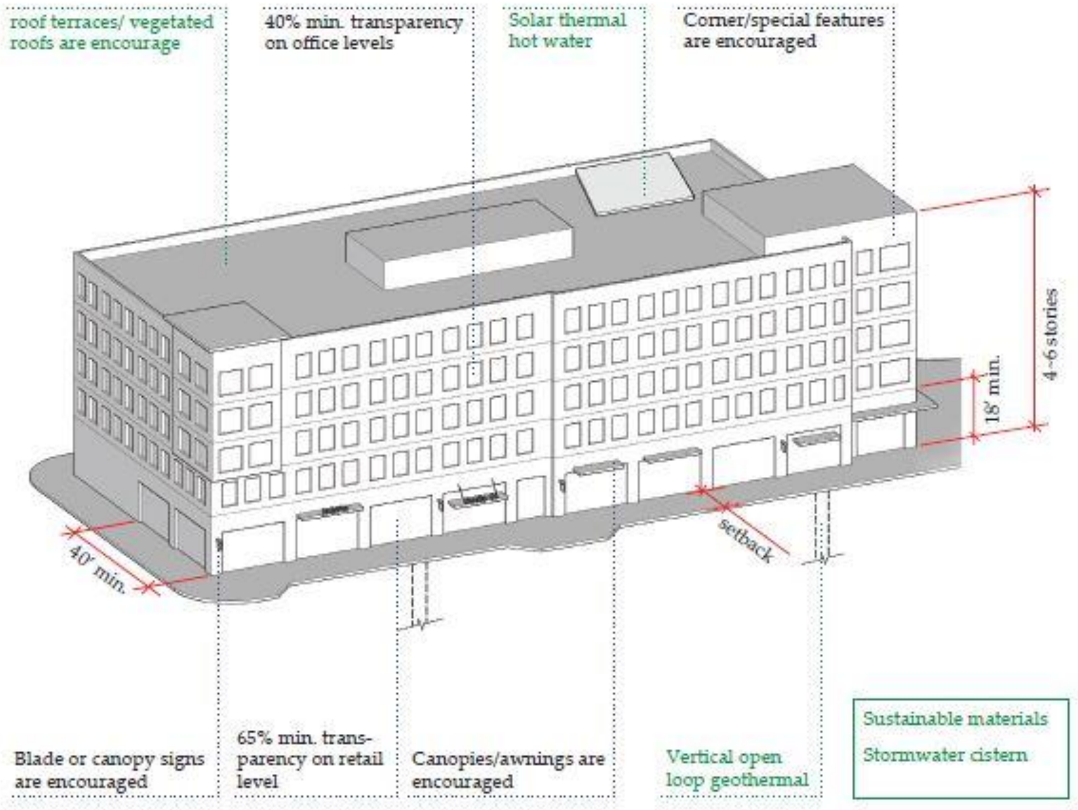


Figure 17: Mixed-use office buildout example, source: (EDAW Inc., 2009)

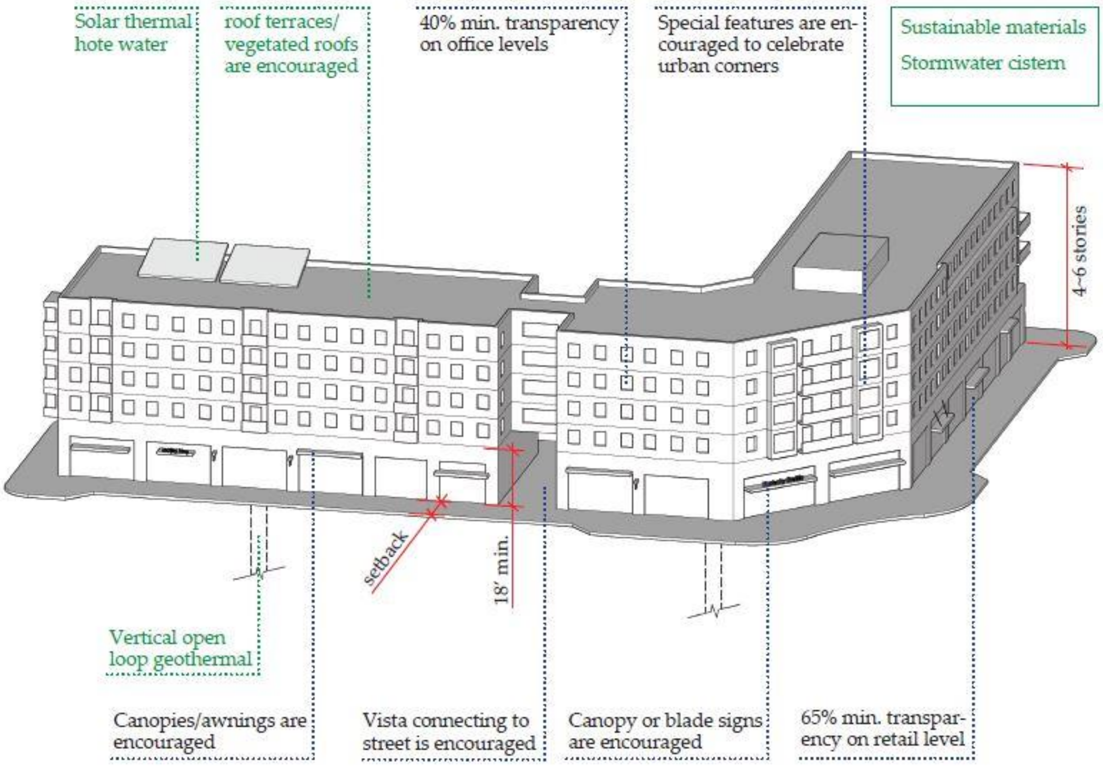


Figure 18: Mixed-use residential buildout example, source: (EDAW Inc., 2009)

One standout building that could be ideal for an active innovation district is a mixed-use hotel building with a retail option on the ground floor. Sitting on a maximum 15,000 sq. ft. footprint with an aligning setback, the ground floor should 15 ft. high with 5-8 additional stories on top which will be 12 ft. high. The ground floor should have a minimum 60% façade transparency and the rest should have a 30% minimum transparency on the building shell.

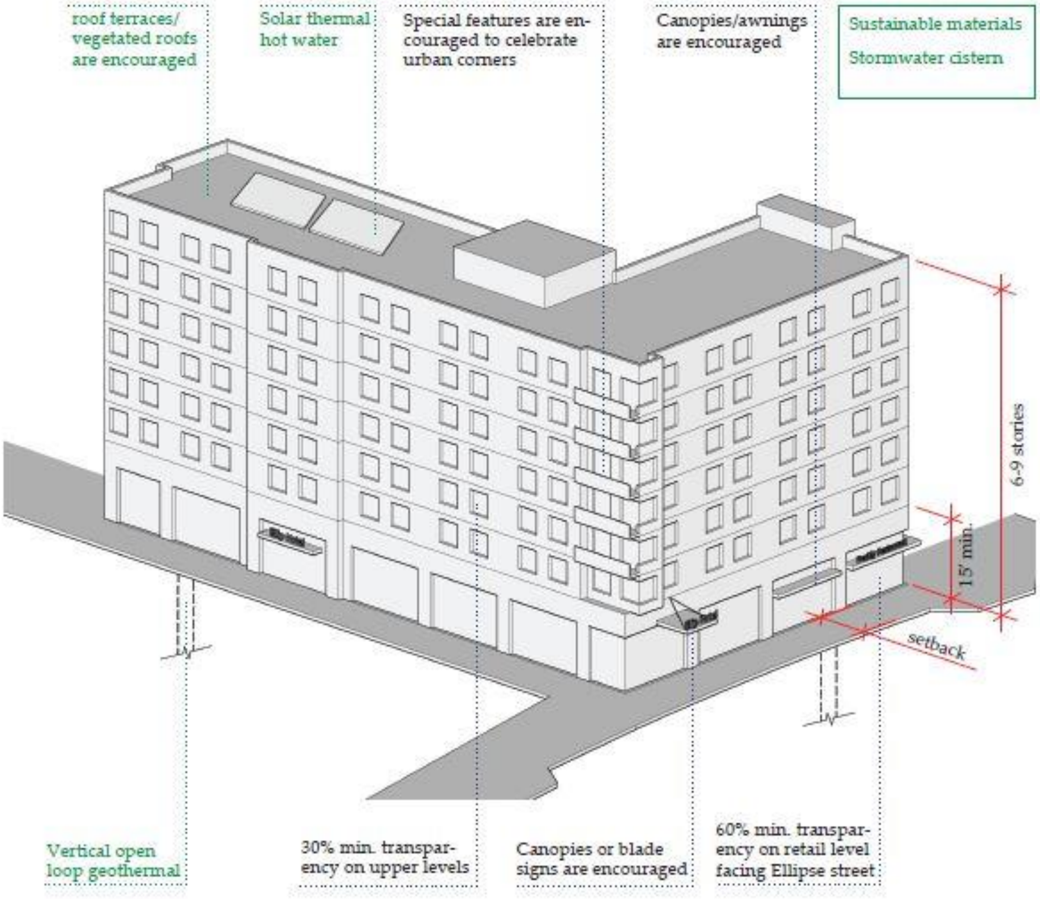


Figure 19: Mixed-use hotel buildout example, source: (EDAW Inc., 2009)

Having a set of physical parameters for each development type that could be assigned to each block formed within the ROWs, all options could be coded in for the automated scenario builder. It is crucial to script the scenario builder in a manner that the user could choose the optional and fluid elements of the design such as building types and number of floors to be able to interact with the scenarios to implement the feedback given in real time. Simplifying and clarifying the various elements within the code is an essential component for the user to be able to work best with scenario builder.

Chapter 5: The scenario builder

After having the basic elements needed for scripting the CGA rules for the scenario builder set up, rules are coded in, refer to Appendix 1 for the script, and the CityEngine interface is set up in a way that would create the most efficient scenario building tool to improve productivity of conversation with clients and stakeholders. To achieve this goal, there are steps that need to be taken before meeting with clients and while having a conversation with the stakeholders.

Before the meeting it's critical to get all the available data for the existing conditions of the site set up in CityEngine. This includes GIS data for the ROWs, parcels of land, building footprints, tree placements, etc. The more data is acquired beforehand, the more accurate the work would be. During the meeting feedback given such as street realignments, new ROWs, building deconstructions, proposed buildings, etc. should be implemented on the go to show the both the physical changes and report the shift in uses or any other reportable data to check each scenario with the general goals of the community. In summation the steps for setting up and utilizing the automated scenario builder is as follows:

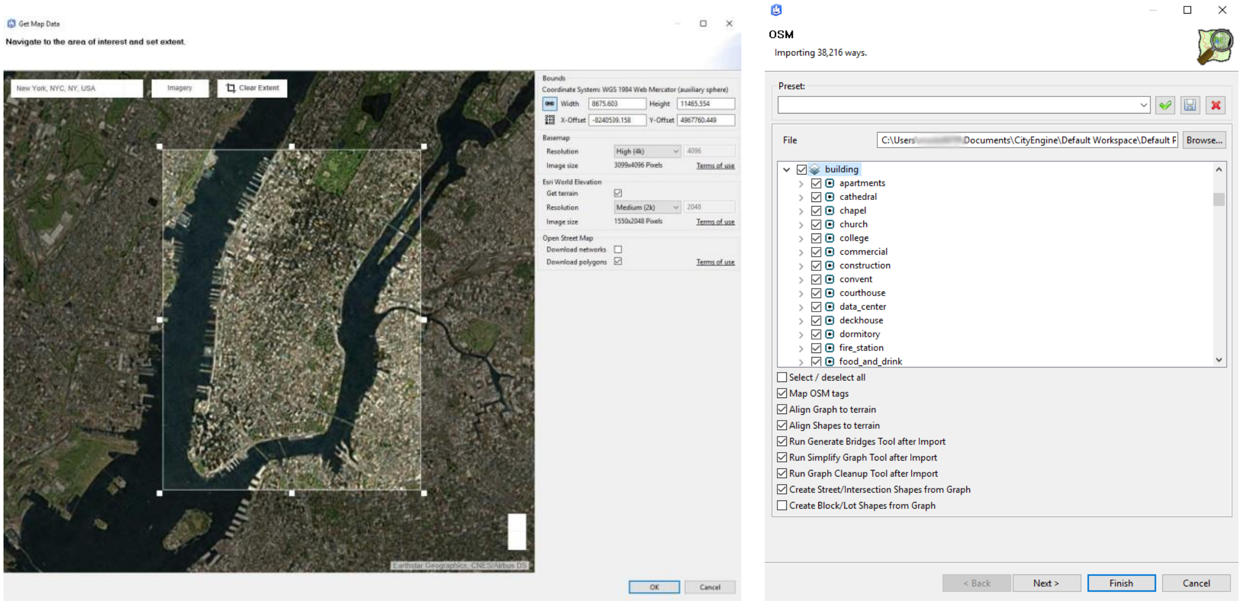
- Setting up existing conditions: Acquiring and embedding existing data for site
- Layering the file: Setting a scenario-based layering system for the file
- Creating street networks: Manually creating the proposed street network for each scenario and assigning the proposed street CGA rule to each ROW
- Developing scenarios: Assigning the desired CGA rule to each parcel/block shaped
- Reporting data: Reporting back aggregated data through the dashboard or report tab of the CityEngine software

Process of using the scenario builder

Setting up existing conditions: Like every planning effort, having a good set of accurate existing data is crucial for the process of working with an automated scenario builder. This will make the efforts of planners and designers more accurate and reliable as well as streamlining the process. Various locations have different data availability, checking with local authorities and clients are crucial, but there are some open source resources to extract some basic data on different locations such as USGS, governmental websites, Open Street Map, etc.

CityEngine has a direct data extraction platform, get map data, from Open Street Map data which is not the most accurate and up-to-date source in many locations but is very helpful in sites with data scarcity. The more layers and accurate data is acquired, the closer to reality the existing conditions will be, but the two crucial components of creating existing conditions are having the street network, CityEngine cannot function without a street network, and building footprint data to know where building are located, in order to create their masses.

For the site in this exercise a 500-acre undeveloped site is chosen with only one existing road neighboring the northern edge to have maximum flexibility for automated development of the site. Also, the site is treated as a hypothetical designated innovation district which will be developing with various uses in the future. Since there's no real-world client, the qualitative and intangible factors affecting development are left out of the process and focus is solely on the physical features of the site. The main physical feature in this case would be topography of the site, as the remaining elements such as blocks, parcels of land, ROWs, etc. are created. ourselves in various hypothetical scenarios.



Navigate to the area of interest and set extent.

Bounds
Coordinate System: WGS 1984 Web Mercator (auxiliary sphere)
Width 8675.603 Height 11465.554
X-Offset -8240539.158 Y-Offset 4967760.449

Basemap
Resolution High (4k) 4096
Image size 3099x4096 Pixels Terms of use

Esri World Elevation
Get terrain ☑
Resolution Medium (2k) 2048
Image size 1550x2048 Pixels Terms of use

Open Street Map
Download networks ☐
Download polygons ☑ Terms of use

Importing 38,216 ways.

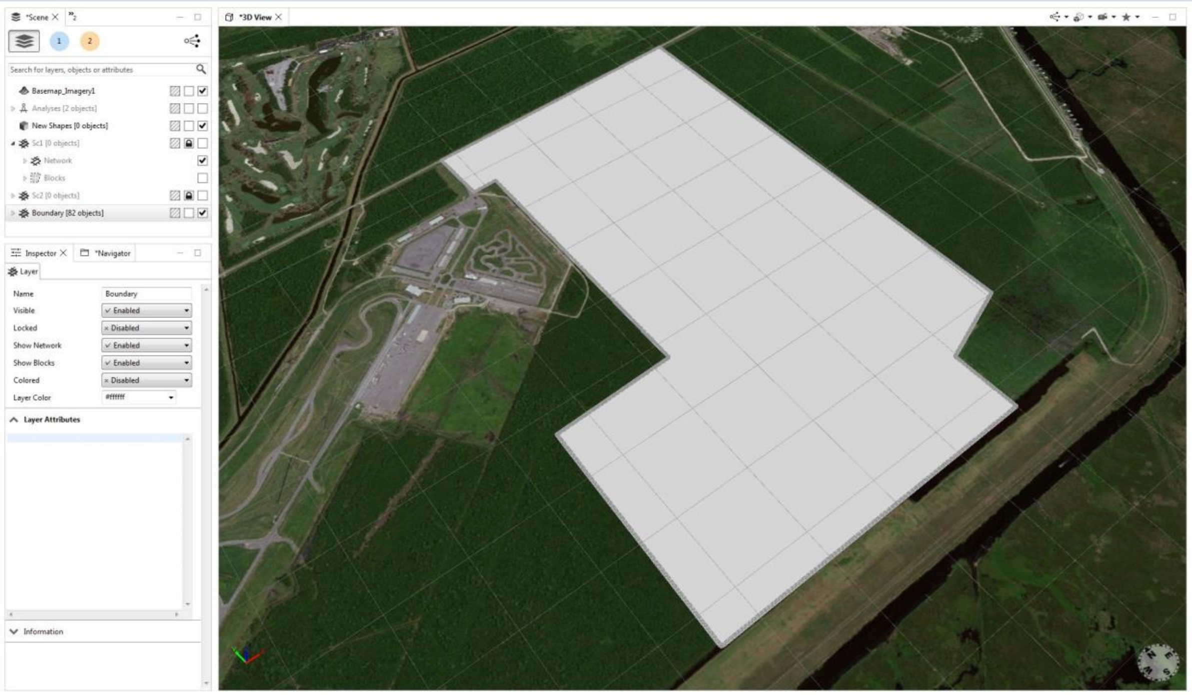
File C:\Users[redacted]Documents\CityEngine\Default Workspace\Default F

- ☑ building
  - ☑ apartments
  - ☑ cathedral
  - ☑ chapel
  - ☑ church
  - ☑ college
  - ☑ commercial
  - ☑ construction
  - ☑ convent
  - ☑ courthouse
  - ☑ data_center
  - ☑ deckhouse
  - ☑ dormitory
  - ☑ fire_station
  - ☑ food_and_drink

☐ Select / deselect all
☑ Map OSM tags
☑ Align Graph to terrain
☑ Align Shapes to terrain
☑ Run Generate Bridges Tool after Import
☑ Run Simplify Graph Tool after Import
☑ Run Graph Cleanup Tool after Import
☑ Create Street/Intersection Shapes from Graph
☐ Create Block/Lot Shapes from Graph

Figure 20: Map data entry in the CityEngine software, source: (Esri R&D Center, 2017)

Layering the file: For being able to work with the software as fast as possible during a meeting with clients, it is crucial to set up the most efficient layering system to be able to toggle between multiple scenarios quickly. The layering set up would vary greatly in different projects as the discussion points could be one specific site in the planning area, multiple districts, or even the entire planning area. It is important to create flexibility to switch between various scenarios in the areas of discussion while avoiding redundancy in the constants within the planning area. In the case of this sample site the layers are set up for the entire planning area because other than the one road to the north of the site, every other element should be a fluid interchangeable component up for discussion with the hypothetical client.



Name: Boundary
Visible: ✓ Enabled
Locked: × Disabled
Show Network: ✓ Enabled
Show Blocks: ✓ Enabled
Colored: × Disabled
Layer Color: #ffffff

Figure 21: Site boundary and layers set up

Creating street networks: One of the main elements necessary for building out the urban environment in CityEngine, is the street network within the area. In this approach towards utilization of procedural modeling, the existing network is set up before engaging in conversation with stakeholders while providing the flexibility to manually test out various alternatives for proposed street networks within the design boundary. One of the benefits of having a manual process is that creation of the street network becomes the first opportunity for interactive feedback built into the process that shapes and is the first step for differentiating various scenarios.

Using the freehand, polygon, and edit streets tools the street network is create and modified while assign subdivision parameters to blocks formed with each network alternative. After creating the ROW centerlines with the tools, parameters needed for the street CGA rule to function correctly are assigned, such as ROW width, lane width, sidewalk width, etc. For better navigation of the model in the next steps, generation of streets is postponed until development types are assigned to each parcel or block.

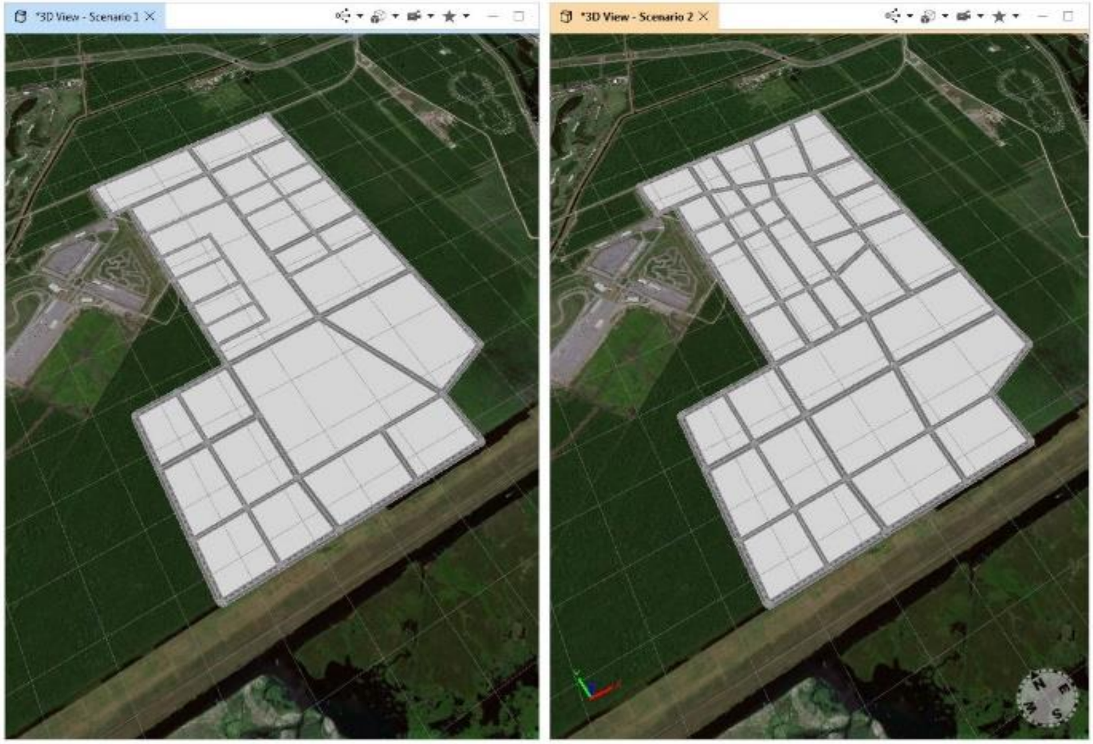
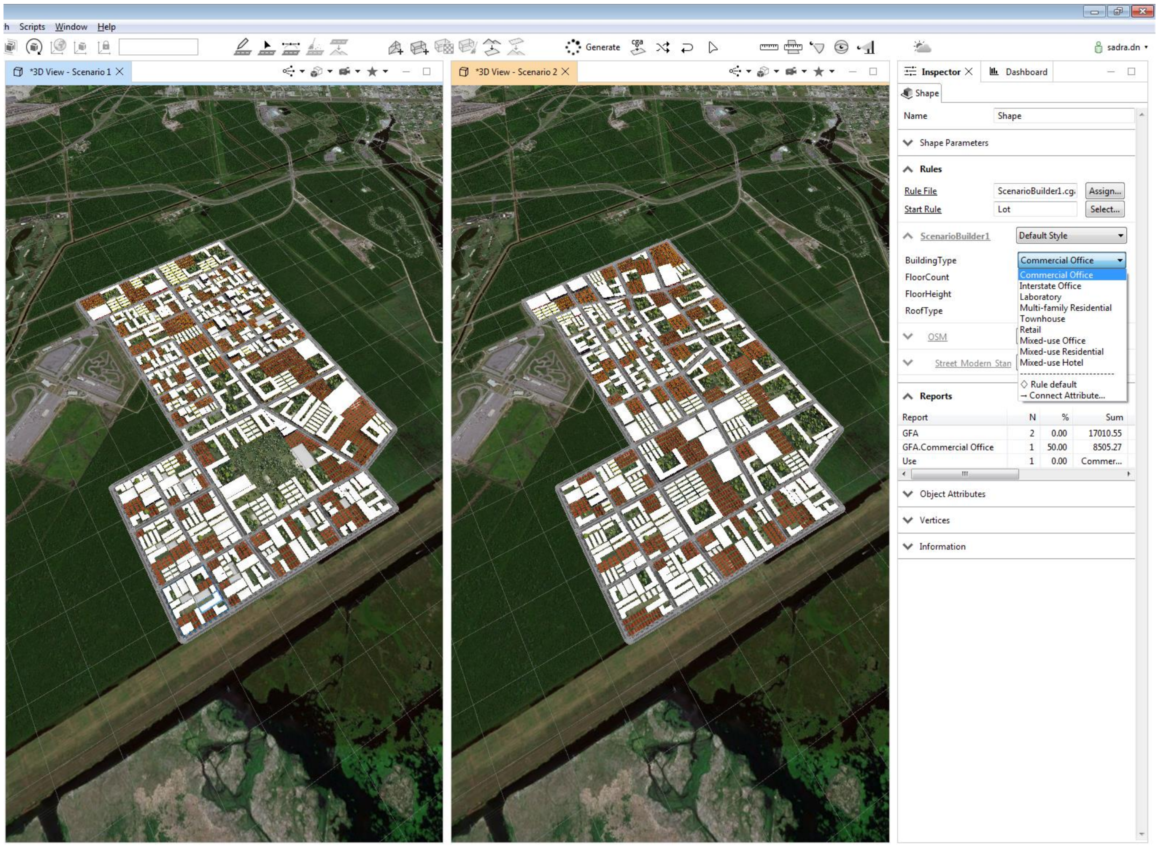


Figure 22: Street network creation for multiple scenarios

Developing scenarios: Having the street network for each scenario set up, consultants can start asking the question of what kind of development should be implemented on each piece of land, whether the shapes created are blocks or parcels. The scenario builder is scripted in a way that after assigning the CGA rule to the shapes provides us a drop-down menu to choose from various development types coded in. This process provides the opportunity to create different options of development types on multiple parts of the planning area quickly during a meeting with clients as opposed to having to take notes of the feedback and implementing it for a future meeting with the manual approach to modeling.



Name: Shape

Rule File: ScenarioBuilder1.cga

Start Rule: Lot

ScenarioBuilder1: Default Style

BuildingType: Commercial Office

Report	N	%	Sum
GFA	2	0.00	17010.55
GFA.Commercial Office	1	50.00	8505.27
Use	1	0.00	Commer...

Figure 23: Assigning development type to each shape in scenarios

Reporting data: After creating models of each scenario based on the feedback from the client or stakeholders, reporting back the overall information for each formed scenario is a crucial part of the planning process because it is important to each see how each scenario meets the goals of the planning effort. In the scenario builder for this exercise only have Gross Square Footage (GSF) and use type are coded into the reporting function, however many more factors could be coded in depending on additional data provided for scenario planning.

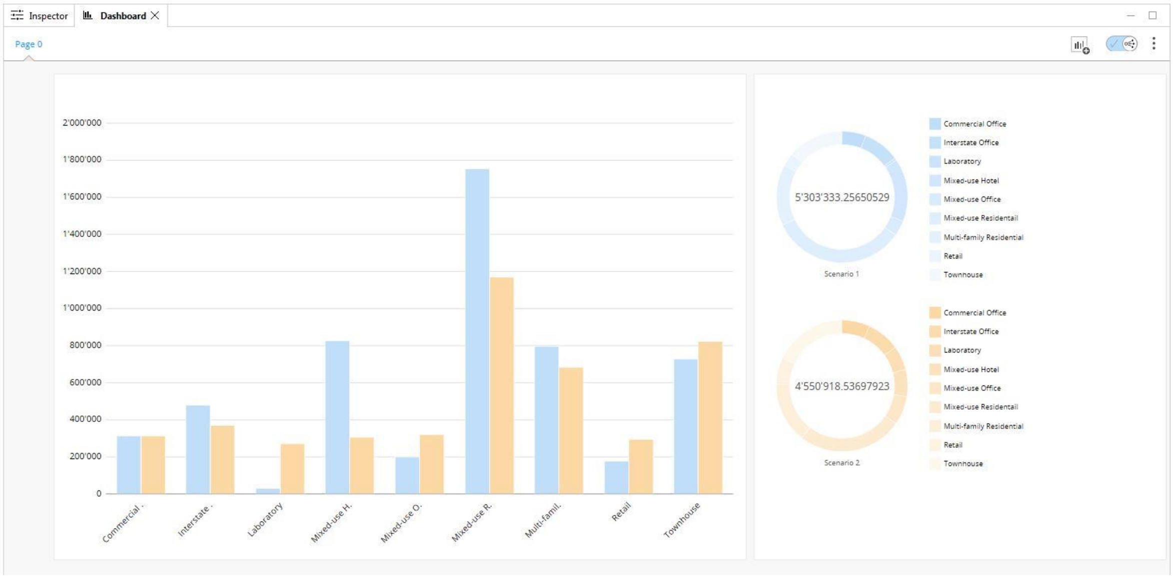


Figure 24: Dashboards as the reporting interface for comparing scenarios

Visualizing the proposal

The main benefit of a computational approach towards modeling alternatives is that it expedites the planning process and streamlines achieving a definitive version/s for a proposal. In addition, the procedural approach provides the opportunities for better visualizing each alternative as it is drafted. CityEngine provides various alternatives for visualization such as, 3d snapshots from various points of view could be taken from the basic models, images with texturing of the massing of each scenario, or a VR experience of the environment are some of the visualization methods built in to the CityEngine software.

Models created in CityEngine could also be exported as Wavefront OBJ, Autodesk FBX, or many other formats common in Visual Effects (VFX). Masses could be imported into gaming engines such as Unreal, CityEngine has a direct export for Unreal, or Unity for high quality visualization techniques such as realistic videos, interactive VR environments, interactive games, etc. This level of visualization is only limited by the imagination and investment on a project, as CityEngine is one of the main software used for modeling urban environments in the movie and gaming industries. Virtual cities in movies such as Independence Day: Resurgence, Zootopia, Blade Runner 2049, etc. or games such as Assassin’s Creed and GTA series have been created in CityEngine and interpreted in various visualization styles which vary greatly from realistic imagery to abstract visualization styles.

The extent of visualization capabilities available to planners/designers while using this procedural approach towards creating an automated scenario builder could be a topic for a future research as various visualization techniques are implemented and weighed against each other in a visual preference survey. It is important for planners and designers to invest in furthering the quality of work within the visualization of their work as it is the language they communicate to the public with.

Chapter 6: Conclusion

Automating the labor-intensive process of modeling numerous elements in an urban environment and reporting back the data for each scenario can greatly accelerate the process. It allows planners/designers to create a larger set of scenarios and spend more time on the quality of deliverables as opposed to manually having to create elements one by one and extracting the data from each component created. Expediting this process to a certain degree would enable us to be able to build out and report back numerous variations of development within a certain planning area on the go, while having a conversation with the clients or stakeholders. This creates an opportunity for instant feedback to be reflected on each alternative and provides the chance for more productive engagement with the planning process.

The strength of a procedural approach is built in expediting the process of repetitive physical operations within modeling and information reporting for various scenarios. However, the limitation of lacking the ability to make actual design decisions is one factor that needs to be kept in mind. It is based on the current limitation of technology that computers can't make complex decision like human beings, advancement of AI might be able to resolve this limitation in the future, and steps for manual human intervention should be built into the automated scenario builder. It is the crossroads of manual intervention and automated processing of recurring functions that can spark engaging conversations while producing built out masses and information about each alternative on the go.

This approach can greatly streamline the process of scenario planning since it creates a generative and dynamic workflow that could adapt to specific needs of each project. This interactive tool can implement feedback through manual editing that can clarify the conversation with production of basic visualization for each scenario in real-time and the reporting interface provides the opportunity for evaluation of each alternative for better decision making.

Challenges of creating the scenario builder

One of the main challenges of this approach as a planner is the shift from the conventional manual method of modeling to an algorithmic one. This requires an alteration in the qualitative approach taken in many planning practices into a solely mathematical and quantitative mind set. Also, having to rely on random numbers instead of making informed decisions for creation of variety leads to having some undesirable results in placing of elements in a fully automated process which need the manual intervention part of the process to be applied for creating the final build out.

In this report, due to limitations of time and resources, only a small number of development types are scripted, and many elements required for better representation of a scenario have been eliminated. With all restrictions in mind, an automated scenario builder is created which could work efficiently for having an interactive engagement process for planning an innovation district. For further advancement of the tool, a library of development typologies could be built; categorized by uses, construction technology, regional context, design brandings, etc. to be able to have the elements built in for all projects planners might encounter.

Professional use of such a tool could be implemented by deploying a survey within designers and planners to create a library of parameters for various development types and scripting all the elements in, with a detailed description such as use by floor, energy usage, employment opportunities, sustainability or resilience factors, etc. for better visualization and reporting of the information embedded with each scenario. This library of various development types could be used as a reference to create tailored automated scenario builders for each project planners encounter, while scripting additional rules suited for the detailed specification of each plan. Employment of this method would lead to creation of an ever-growing library of development typologies to automate as many procedures as possible within the planning process, while having the fluidity to adapt to certain requirements and individualities of each project.

Computational tools in the planning field have been greatly advancing in the past few years and been assisting planners/designers process larger sets of data in shorter time periods. The advancement of these tools is dependent on innovation in the fields of technologies building these tools and finding creative ways of utilizing them. It is important to remember that computational planning is not a replacement for the role of the planners but an advanced tool that can greatly further their work and the more tools are available, the more equipped consultants will be for handling complex wicked problems.

The lack of decision making within the scripts could be resolve in the future with application of machine learning within the codes utilizing the Python scripting capability of the software. Applying AI within the rule packages, we might be able to create conditions that the rule, through machine learning, can evolve as each iteration of the rule is applied and altered. This could lead to the possibility of having a code that is aware of some especial circumstances and chooses to break the rule in those especial cases. This could result into being capable of coding rules in the future that can make design-based decisions and maybe one day automating the main portion of the planning process.

Appendix

Scenario builder CGA rule:

```
/**
 * File:    ScenarioBuilder.cga
 * Created: 24 Apr 2018 14:10:10 GMT
 * Author:  sd33677
 */

version "2017.1"

import OSM:"rules/TestInt1.cga"

@Range ("Flat", "Gable")
attr RoofType =

        case BuildingType == "Commercial Office" || BuildingType == "Interstate
Office" ||
        BuildingType == "Laboratory" ||
        BuildingType == "Mixed-use Office" || BuildingType == "Mixed-use
Residential" || BuildingType == "Mixed-use

Hotel" : "Flat"
        case BuildingType == "Townhouse" : "Gable"
        else : randomRoofType

randomRoofType =
        case scope.sx > 20 || scope.sy>20 : "Flat"
        case p(0.5): "Gable"
        else: "Flat"

@Range ("Commercial Office", "Interstate Office", "Laboratory", "Multi-family
Residential",
"Townhouse", "Retail", "Mixed-use Office", "Mixed-use Residential", "Mixed-use
Hotel")
attr BuildingType = randomBuilding


randomBuilding =
        15%: "Commercial Office"
        5%: "Interstate Office"
        2%: "Laboratory"
```

```
    10%: "Multi-family Residential"
    20%: "Townhouse"
    5%: "Retail"
    10%: "Mixed-use Office"
    3%: "Mixed-use Hotel"
    else: "Mixed-use Residential"



########################################FEATURE
HEIGHT####################################

@Range (5,25)
attr FloorHeight =
    case BuildingType == "Commercial Office" : 4 #1-3floors (12 ft.)
    case BuildingType == "Interstate Office" : 4.5 #2-8floors (14 ft.)
    case BuildingType == "Laboratory" : 6 #1-3floors (20 ft.)
    case BuildingType == "Multi-family Residential" : 3 #3-5floors (10 ft.)
    case BuildingType == "Townhouse" : 3 #2-3floors (10 ft.)
    case BuildingType == "Retail" : 7 #?floors (22-35 ft.)
    case BuildingType == "Mixed-use Office" : 4 #18 ft. pilot + 3-5floors (12 ft.)
    case BuildingType == "Mixed-use Residential" : 3 #18 ft. pilot + 3-5floors (10 ft.)
    case BuildingType == "Mixed-use Hotel" : 4 #15 ft. pilot + 5-8floors (12 ft.)
    else : 50

@Range (5,25)
attr FloorCount =
    case BuildingType == "Commercial Office" : rint(rand(1,3)) #1-3floors
    case BuildingType == "Interstate Office" : rint(rand(2,8))#2-8floors
    case BuildingType == "Laboratory" : rint(rand(1,3)) #1-3floors
    case BuildingType == "Multi-family Residential" : rint(rand(3,5)) #3-5floors
    case BuildingType == "Townhouse" : rint(rand(2,3)) #2-3floors
    case BuildingType == "Retail" : rint(rand(1,2))#?floors
    case BuildingType == "Mixed-use Office" : rint(rand(4,6)) #18 ft. pilot + 3-
5floors
    case BuildingType == "Mixed-use Residential" : rint(rand(4,6)) #18 ft. pilot + 3-
5floors
    case BuildingType == "Mixed-use Hotel" : rint(rand(6,9)) #15 ft. pilot + 5-8floors
    else : 50
```

```
#####################################RULES TO
BUILD##################################

@StaRtrule
Lot-->
        OSM.GreenGround
        Lot1
Lot1-->
        case BuildingType == "Commercial Office" :  #Footprint:~70*30
                offset(-3, inside)
                shapeL(30,30) { shape : Build report("GFA.Commercial Office",
FloorCount*geometry.area) | remainder:

offset(-10,inside) OSM.FewTrees }
        case BuildingType == "Interstate Office" : #Footprint:~120*30
                offset(-3, inside)
        shapeU(30,30,40) { shape : Build report("GFA.Interstate Office",
FloorCount*geometry.area) | remainder: offset(-

10,inside) OSM.FewTrees }
        case BuildingType == "Laboratory" : #Footprint:~80*40
                split (x) { ~3:OSM.GreenGround | ~80:
                        split(z) { ~7:OSM.GreenGround| ~40: Build
report("GFA.Laboratory", FloorCount*geometry.area) |

~3:OSM.GreenGround}
                | ~3:OSM.GreenGround}
        case BuildingType == "Multi-family Residential" : #Footprint:~60*20
                split (x) { ~3: OSM.GreenGround | ~60:
                        split(z) { ~7: OSM.GreenGround| ~20: Build report("GFA.Multi-
family Residential",

FloorCount*geometry.area) | ~3: OSM.GreenGround}*
                | ~3: OSM.GreenGround}*
        case BuildingType == "Townhouse" : #Footprint:~10*20
                split (x) { ~3: OSM.GreenGround | ~10:
                        split(z) { ~7: OSM.GreenGround| ~20: Build
report("GFA.Townhouse", FloorCount*geometry.area) |

~3:OSM.GreenGround}*
                | ~3:OSM.GreenGround}*
        case BuildingType == "Retail" : #Footprint:~120*80
                split (x) { ~3:OSM.GreenGround | ~120:
```

```
            split(z) { ~7:OSM.GreenGround| ~80: Build report("GFA.Retail",
FloorCount*geometry.area) |

~3:OSM.GreenGround}
        | ~3:OSM.GreenGround}
    case BuildingType == "Mixed-use Office" : #Footprint:~100*25
        offset(-3, inside)
        shapeL(25,25) { shape : Build report("GFA.Mixed-use Office",
FloorCount*geometry.area) | remainder:

offset(-10,inside) OSM.FewTrees}
    case BuildingType == "Mixed-use Residential" : #Footprint:~120*25
        offset(-3, inside)
        shapeL(25,25) { shape : Build report("GFA.Mixed-use Residentail",
FloorCount*geometry.area)| remainder:

offset(-10,inside) OSM.FewTrees }
    case BuildingType == "Mixed-use Hotel" : #Footprint:~150*35
        offset(-3, inside)
        shapeO(35,35,25,25) { shape : Build report("GFA.Mixed-use Hotel",
FloorCount*geometry.area) | remainder:

offset(-10,inside) OSM.FewTrees }
    else : Building.

Build-->
    report("GFA", FloorCount*geometry.area)
     report("Use" , BuildingType)
     extrude(FloorHeight*FloorCount)
     comp (f) {top:Roof | side:SplitWall}

Roof-->
    case RoofType == "Flat" : extrude (0.5)
    case RoofType == "Gable" : GableRoof
    else : color (1,0,0)

GableRoof-->
    roofGable(30, 0.5, 0.5)
    color(.65,.25,.128)

#Need to code in wall formations based on urban design elements
```

```
#####################################WALL RULES#################################

SplitWall-->
        split(y) {FloorHeight:ColorWall}*


ColorWall-->
        case BuildingType == "Commercial Office" || BuildingType == "Interstate Office" :
                color (.218,.102,.210)
                OWall
        case BuildingType == "Laboratory" :
                color (.238,.130,.238)
                LabWall
        case BuildingType == "Multi-family Residential" || BuildingType == "Townhouse" :
                color (255,255,0)
                ResWall
        case BuildingType == "Retail" :
                color (1,0,0)
                RetWall
        case BuildingType == "Mixed-use Office" || BuildingType == "Mixed-use Residential"
                || BuildingType == "Mixed-use Hotel" :
                color(.90,.69,.255)
                MixWall
        else: finalshape.

OWall-->
        case split.index == 0:
                DoorWall
        else:
                Facade50

LabWall-->
        case split.index == 0:
                DoorWall
        else:
                Facade25

ResWall-->
```

```
    case split.index == 0:
        DoorWall
    else:
        Facade40

RetWall-->
    case split.index == 0:
        DoorWall
    else:
        Facade75

MixWall-->
    case split.index == 0:
        color(1,0,0)
        DoorWall
    else:
        Facade50

############################URBAN DESIGN
FEATURES##############################

DoorWall-->
    split(x) { ~1:X.| 2:WindowSplitY | ~1.5:X.|  1.2: DoorSplitY| ~1.5:X. |
2:WindowSplitY| ~3:X.}*
        extrude(-1)
DoorSplitY-->
    split(y) { 3:Door| ~1:X. }

Door -->
    offset(-0.2)
    comp(f) {inside: color(0,0,0) X.| border: color(1,1,1) X. }

WindowSplitY-->
    split (y) { '1/3:X.| '1/3:Window | '1/3:X.}

Window-->
    offset(-0.1)
    comp(f) {inside: color(0,0,0) X.| border: color(1,1,1) X. }

Facade25-->
    50%: split(x) { ~1:X.| 2:WindowUPSplitY | ~2:X.| 2:WindowUPSplitY| ~1:X.}*
    25%: split(x) { ~1:X.| 1:WindowUPSplitY | ~1:X.| 1:WindowUPSplitY| ~1:X.|
1:WindowUPSplitY| ~1:X.}*
```

```
        else: split(x) { ~1:X.| 0.9:WindowUPSplitY | ~ 0.2:X.| 0.9:WindowUPSplitY | ~
2:X.| 0.9:WindowUPSplitY | ~

0.2:X.| 0.9:WindowUPSplitY |~1:X.}*

Facade40-->
        50%: split(x) { ~1:X.| 2:WindowUPSplitY | ~1:X.| 2:WindowUPSplitY| ~1:X.}*
        25%: split(x) { ~1:X.| 1:WindowUPSplitY | ~0.5:X.| 1:WindowUPSplitY|
~0.5:X.| 1:WindowUPSplitY| ~1:X.}*
        else: split(x) { ~1:X.| 0.9:WindowUPSplitY | ~ 0.2:X.| 0.9:WindowUPSplitY | ~
1:X.| 0.9:WindowUPSplitY | ~

0.2:X.| 0.9:WindowUPSplitY |~1:X.}*

WindowUPSplitY-->
        split (y) { '1/4:X.| '1/2:Window | '1/4:X.}

Facade50-->
        50%: split(x) { ~1:X.| 2:WindowUpSplitY | ~2:X.| 2:WindowUpSplitY| ~1:X.}*
        25%: split(x) { ~1:X.| 1:WindowUpSplitY | ~1:X.| 1:WindowUpSplitY| ~1:X.|
1:WindowUpSplitY| ~1:X.}*
        else: split(x) { ~1:X.| 0.9:WindowUpSplitY | ~ 0.2:X.| 0.9:WindowUpSplitY | ~
2:X.| 0.9:WindowUpSplitY | ~

0.2:X.| 0.9:WindowUpSplitY |~1:X.}*

Facade75-->
        50%: split(x) { ~1:X.| 2:WindowUpSplitY | ~1:X.| 2:WindowUpSplitY| ~1:X.}*
        25%: split(x) { ~1:X.| 1:WindowUpSplitY | ~0.5:X.| 1:WindowUpSplitY| ~0.5:X.|
1:WindowUpSplitY| ~1:X.}*
        else: split(x) { ~1:X.| 0.9:WindowUpSplitY | ~ 0.2:X.| 0.9:WindowUpSplitY | ~
1:X.| 0.9:WindowUpSplitY | ~

0.2:X.| 0.9:WindowUpSplitY |~1:X.}*

WindowUpSplitY-->
        split (y) { '1/8:X.| '3/4:Window | '1/8:X.}
```

Works Cited

Avin, U. P., & Dembner, J. L. (2001, November). Geeting scenario-building right. Planning, 22-27.

Bosselman, P. (1998). Representation of places: Reality and realism in city design. Berkeley: University of California Press.

Bruchell, R. W., Downs, A., McCann, B., & Mukhreji, S. (2005). Sprawl costs: Economic impacts of unchecked development. Washington, DC: Island Press.

Christopher, A., Ishikawa, S., & Silverstone, M. (1977). A pattern language: Towns, buildings, construction. New York: Oxford University Press.

Costanza, R. (1999). Four visions of the century ahead: Will it be Star Trek, Ecotopia, Big Government or Mad Max? Futurist, 33, No. 2, 23-30.

EDAW Inc. (2009). Coldstream research campus design guidelines. University of Kentucky .

Esri R&D Center. (2017). CityEngine Help. Zurich.

Gobal Business Network. (2004). The valley futures project. Retrieved from www.valleyfutures.org

Heijden, K. v. (1996). The art of strategic conversation. New York: John Wiley & Sons.

Hopkins, L. D., & Zapata, M. A. (2007). Engaging the Future. Cambridge, Massachusetts: Lincoln Institute of Land Policy.

Kwartler, M., & Longo, G. (2008). Visioning and Visualization: people, pixels, and plans. Cambridge, Massachusetts: Lincoln Institute of Land Policy.

Maantay, J., & Ziegler, J. (2006). GIS for the urban environment. Redlands, California: ESRI Press.

McLuhan, M. (1967). The medium is the message: An inventory of effects. New York: Bantam.

Muller, P., & Parish, Y. I. (2001). Procedural Modeling of Cities. 28th annual conference on Computer graphics and interactive techniques (pp. 301-308). New York: ACM.

Perkins+Will. (2018). PlanMetrics Guide. On file with author.

Schwartz, P. (1991). The art of the long view. New York: Currency Doubleday.

Talton, J. O., Lou, Y., Lesser, S., Duke, J., Mech, R., & Koltun, V. (2011). Metropolis procedural modeling. ACM Transactions on Graphics (TOG), 11:1-11:14.