PREPARATORY MODULE FOR THE OBJECT-ORIENTED MODELING OF URBAN TERRAIN HYDROLOGICAL RESPONSE

Jusuf J. S. Pah[1] (yuserpbdaniel@yahoo.co.id)

ABSTRACT

This research aimed at preparing an object-oriented module to use in conjunction with an object-oriented urban terrain hydrological response model. The module serves to capture terrain data and prepare them and make them fit as input to the core part of the urban terrain model. This research had constructed the module. It engages five classes of object, which give it capability to faithfully capturing terrain hydrological features and other relevant data, and arrange them in such a way as to serve as fit input to the core part of the urban terrain model. The module represents an urban terrain by meshing it unto a network of rectangle object elements. In order to overcome edge problems associated with the meshing, utilizing smaller size of elements (hence higher number of elements) and weighted average in terms of 'within a cell contained terrain-sheds areas' were advised. It also strongly encouraged the invention of a more automate ways of terrain data capturing, to replace the manual data entry procedure still utilized in this module.

Key Words:
Edge problems; hydrological response; object-oriented approach; preparatory module; terrain-sheds; urban terrain model

INTRODUCTION

Since its invention in 1960 (Poo, Kiong, & Ashok, 2008), the Object-Oriented Approach (OOA) has been widely utilized as a computer programming paradigm in modeling various engineering systems. Wide use of the approach may be attributed to its tremendous capability to mimic the real system that is being modeled. Some notable efforts in this line that can be mentioned here are Struthers & Illidge, (1995) who evaluated the liquid spill potential and loss prevention by an object-oriented tool, Wong & Yong, (1999) who used it to model container terminal analysis, and Sydelco, Hlohowskyj, Majerus, Christiansen, & Dolph, (2001) who developed an object-oriented framework for dynamic ecosystem modeling. Dia (2001) used it in combination with artificial neural network for short-term traffic forecasting, and Weidl, Madsen, & Israelson, (2005) applied the paradigm on root-cause analysis and decision support on the operation of complex continuous processes. Later on Pah, (2011) who recently used the approach to develop a new way of generating moment curvature relationship of a beam section loaded in flexure declared that the approach is much efficient in dealing with multi-material beam cross-section and can be even developed as to be capable of analyzing complex composite bi-axial bending beams. In addition to that, he most recently, published a book in which he presents his works in 2008 of using the object-oriented approach to automate the Water Safety Plans for risk analysis of urban water supply system (Pah, 2013). Tremendous capability of the object-oriented approach lays in its principle concepts: objects, encapsulation, class and inheritance that enable modeling by this paradigm to faithfully replicate the real system that is being represented. Possessing those concepts, OOA stands as the fittest computer programming paradigm for the modeling of engineering systems.

If we look closely at the civil-engineering, one of its fields that can benefit much from the potency of OOA is the urban infrastructure engineering. This may be attributed to attractiveness

1. Department of Civil Engineering, Faculty of Science and Engineering, University of Nusa Cendana – Kupang

of terrain-based modeling (Tucker, Lancaster, Gasparini, Bras, & Rybarczyk, 2001) which is enhanced by the object-oriented paradigm. Modeling hydrological response of an urban terrain in object-oriented paradigm e.g., can serve as an ideal platform by which urban drainage systems as well as its water-supply-system are designed, potency of inundation by surface runoff can be predicted beforehand and preventive measures can be introduced in response to it. Also, a progressive master plan for an urban area can be more appropriately developed, as various future scenarios of terrain development are inputted into the model and studies of their future impact on hydrological response of the terrain are carried out. OOA therefore is an appropriate paradigm for modeling terrain hydrological activity in urban infrastructure engineering.

Modeling an urban terrain in object-oriented paradigm presents two major works: (1) developing an object-oriented module for the registration and preparation of terrain data, be they hydrological, topographical, geological, or demographical etc, and (2) developing an object oriented model that mimics the hydrological dynamic activities of the urban terrain in question. The former is preparatory in nature while the later stands as the main part or the core of the modeling. While the later does not present many difficulties to an object-oriented modeler, the former does. Difficulties in relation with the former arise from the fact that a terrain needs to be meshed into a finite number of interconnected elements, conceptually called object, before further modeling process be carried out. Each of the elements should be able to faithfully replicate features of part of the terrain that it represents. It can be easily conceived that when available terrain data are sparser than the network of elements, as is often the case, faithfulness of the replication is consequentially compromised. This is implied by Mobley, Caroll, & Canann, (1998?). Another challenge which is also in relation with the former is how to process terrain raw data in an object-oriented way, and prepare them as fit input to the core part of the model.

This research dealt with the challenges just mentioned above and seek to develop an object-oriented framework for the preparation of data that will later serve as input into the core part of the object-oriented hydrological terrain model. The framework will come in the form of a module whose principle task is to register terrain data and make it fit for further manipulation in the core part of the model. It limits its scope at a two dimensional terrain in the form of a square (width of the terrain equals to length of the same) and a network of cell-element in the form of a square. It expects an available terrain data which is not sparser and not less in detail then those of the network of cell-elements. At condition where available data are sparser or less in detail than the cell-element network, means of data normalization will be provided. We shall begin the discussion concerning the module by bringing forward the way it is constituted.

CONSTITUTION OF THE MODULE

Since the module is object-oriented in character, it is sensible to begin by presenting the classes of objects that form the module, and only secondly present the way by which the module operates, after all classes of object have been thoroughly explained.

Classes of Object

In accordance with the philosophy of the object-oriented modeling, an object in this module is a conceptual part or a conceptual component of an urban terrain that has a clearly defined function and plays a certain role. An urban terrain with all its associative components will be regenerated and presented in the form of a model when a definite system of objects works together for the terrain in question. Objects of similar kind form a class of object. Followings are classes of object that form this module. One thing needs mention before progressing any further namely that the development of classes of objects elaborated in the following presupposes the existence of a human modeler and a data transcript of the terrain in question.

Class: Preliminary Data Organizer

First class of object in this module is the class: Preliminary Data Organizer. This class possesses only one object namely the Preliminary Data Organizer (PDO). At the commencement of the module, a PDO will be activated and ask for input of the followings:

(i) Sides (S) of the terrain, and

(ii) Number of elements (n_e), unto which the terrain is to be meshed.

After the input of those above, the object calculates length of the side (s) of the elements by the following equation:

$$s = \frac{S}{\sqrt{n_e} - 1} \tag{1}$$

where

s is length of side of the element
S is length of side of the terrain, and
n_e is number of elements,

and Number of Elements within a Row (east-west direction) (NOE_R), as well of Number of Elements within a Column (south-north direction) (NOE_C), in the following way

$$NOE_R = NOE_C = \sqrt{n_e} \tag{2}$$

and creates a relevant Cartesian plane, whose length of x axis and that of y axis are respectively equal to S.

Immediately following this, the PDO generates the object of Data-Cell (DC) in numbers equal to that of n_e and assigns to each of the cell so generated a unique initial by the following algorithm:

1. Unto the first cell is assigned initial mn of which m = 1 and n = 1, side of cell s, and position the cell at coordinate (0;0) in the Cartesian plan;
2. The PDO then will find out whether the number of DCs that has just been assigned equals to any folds of NOE_R. At inequality, the PDO will
   a. increase m by 1 and assigns initial mn to the next cell, side of cell s, and position the cell at coordinate $(0+(m-1)(s);0)$; $\{m \mid 0 < m \le NOE_R;$ m is a round number$\}$ in the Cartesian plan. Following this assignment, the PDO will check if the number of DC that has been assigned has equaled to any folds of NOE_R.

      At inequality, the PDO will pick the next cell, increase m by 1, initial the cell with mn and proceeds by the assigning of side of cell s and positioning of the cell in the same manner as described in 2a above.

   At the equality, the PDO will find out whether the number of DC that has been so far assigned equal to any folds of NOE_C. At inequality, the PDO will pick the next DC,

   b. sets the value m equals to 1, increase n by 1 and assign initial mn to the cell, side of cell s, and position the cell at coordinate $(0+(m-1)(s);0+(n-1)(s))$ $\{n \mid 0 < n \le NOE_C;$ n is a round number$\}$ in the Cartesian plan. The PDO will then proceed in the same manner as described in 2a above, this time with the current value of m is kept static until the number of the assigned cell equal to any folds of NOE_R.

   At equality to the folds of NOE_C, PDO ceases from assigning DCs.

Class: Data-Cells

It is already implied in the foregoing that there exists, in connection with PDO, another class of object in this module which is called class: Data-Cell. A Data-Cell (CD) is an object of the class whose main function is registering features and data of a part of the terrain that it represents. At the event of data registration, each of data-cells will be activated one by one, starting with the one with the lowest initial until the one with the highest. At activation, the cell will register its coordinate of position and its initials as its attributes and will proceed with asking the modeler to input of the following data:

(i) elevation (z);

(ii) intensity (in terms of height) of precipitation (p);

(iii)surface' runoff coefficient[1] (C), and

(iv)surface coefficient of smoothness (n_d), and then

registers them as its attributes.

It is possible that for each of the items above, there exists more than one number of data. At such a condition, the cell will average those values and registers the averaged value as its attribute. Then by sending its attributes to the object of Terrain-Cell-Publisher (explained later), a data-cell concludes its activity.

Class: Terrain Cell Generator

A Terrain Cell Generator (TCG) is the only object in the class: Terrain Cell Generator. When set into function, the object generates the object of Terrain-Cells (TC) (explained in the following) in a number equals to the number of data-cells already generated by the PDO. One TC is paired with one DC. Unto each TC is assigned attributes of its corresponding DC. It can be perceived that when the TCG concludes its task, there exist in the module, Terrain-Cells in number equals to the number of Data-Cells already in existence, a cell of which possesses attributes which are identical with attributes of its corresponding Data-Cell.

Class: Terrain Cell

A Terrain_Cell (TC) is an object in the class: Terrain-Cell that mimics the features and the predicates of a part of an urban terrain that it represents. A graphical representation showing the general structure of this object appears in Appendix 1. It is obvious form the representation that a Terrain_Cell possesses attributes that corresponds to a Data-Cell and various methods which enable it to function as a part of an urban terrain. If other classes are only property of this preparatory module, class Terrain Cell belongs to both the preparatory module and the core part of the model. In this way it serves the functions of a port (or a "socket") through which the module is connected to the Core Part.

Class: Terrain Cell Publisher

This class is formed by a single object: Terrain-Cell-Publisher (TCP). The existence of this object has been implied in the preceding, when Data-Cells was discussed. A TCP displays information sent to it in a Graphic User Interface (GUI) platform, in such a way as to give areal picture to the users on features of the terrain in question and its distribution throughout the same.

[1] Known otherwise as Coefficient of Imperviousness (C)

Operation of Objects

As classes of object have been explained in the preceding, it is proper now to discuss how the module with all its objects operates. For that purpose and to give a picture of how numbers are manipulated in the operation, a sample terrain whose hydrological and topographical data shown in Figure 1 is proposed.

It can be seen in the figure that the terrain is a 30 m side rectangle. There exist a multi-unit residential area at the north-western part of the terrain (violet in color), with elevation of +10 m and runoff coefficient, C, of 0.7. Most of the northern part of the terrain comprises of play ground (blue in color) with runoff coefficient of C = 0.3, at elevation of +7 m. The remaining part of the terrain is a bituminous pavement whose elevation is + 5 m, runoff coefficient, C = 0.8). All runoff coefficients used in this sample were adopted from (Akinyemi, 2006). Constant and uniform precipitation of 100 mm/min is assumed for the whole terrain. Surface coefficient of smoothness in the value of 0.02 has also been assumed for every part of the terrain. It is important to remind here that the operation of the module presupposes the existence of a human modeler.

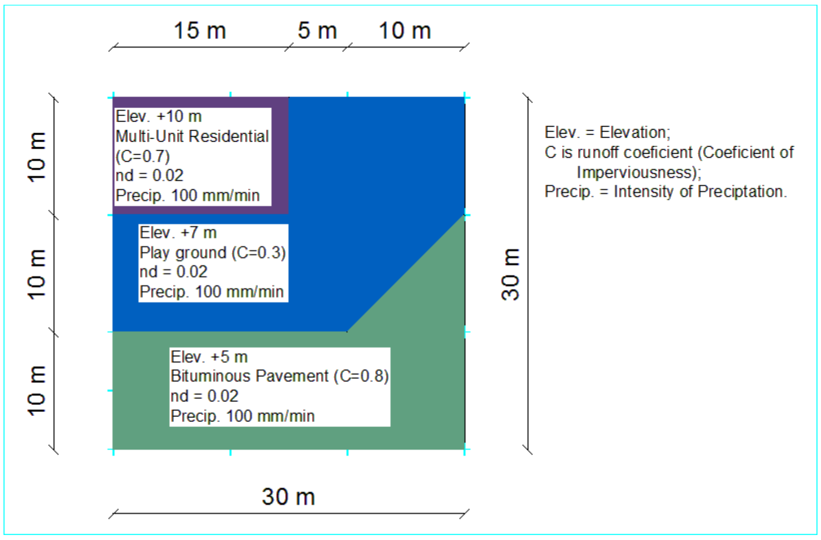


Figure 1. Graphical Data Transcript of the Sample Terrain

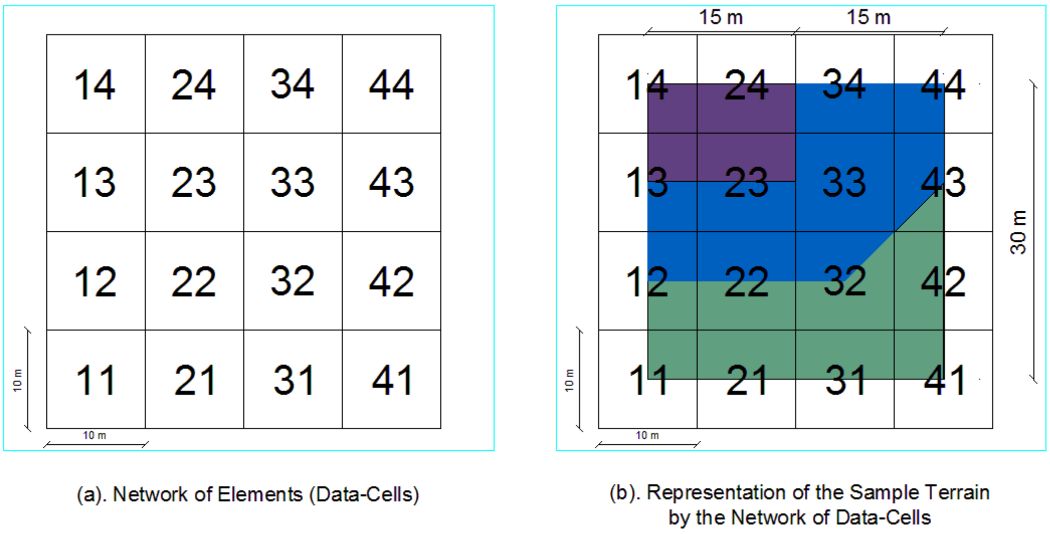


Figure 2. Representation of the Sample Terrain by the Mesh of Data-Cells.

The operation starts by the activation of the PDO by the modeler. For the time being, the terrain will be meshed into a network of 16 cell elements (Figure 2). As side of terrain S = 30 m, and number of elements $n_e = 16$ are inputted, the PDO calculates length of side of the element by way of equation (1), resulting in

$$s = \frac{30}{\sqrt{16}-1} = 10 \text{ m} \tag{3}$$

Length of side of each of the element therefore is 10 m. Following this the PDO determines the NOE_R and NOE_C by way of equation (2) which results in:

$$NOE_R = NOE_C = \sqrt{16} = 4 \tag{4}$$

and creates a relevant Cartesian plan, and 16 DCs. Unto each of the DC is assigned an initial and is positioned in a coordinate by way of the algorithm described previously under “class: Preliminary Data Organizer”, resulting in the generation of network of DCs shown in Figure 2(a), each of which represents a part of the terrain as explained by Figure 2(b).

Each of the DC is then summoned into action one by one, starting from the one with the lowest initial (DC 11) until the one with the highest (DC 44). At every session, the modeler will have to input: z, C n_d and p, as described previously, into the active cell. Into the cell DC 11, e.g. shall be inputted: z = 5 m; C = 0.8, nd = 0.02 and p = 100 mm/min, into cell DC 14 shall be inputted: z = 5 m; C = 0.7, nd = 0.02 and p = 100 mm/min, into cell DC 21 shall be inputted: z = 5 m; C = 0.8, nd = 0.02 and p = 100 mm/min, into cell DC 24 shall be inputted: z = 5 m; C = 0.7, nd = 0.02 and p = 100 mm/min, into cell DC 31 shall be inputted: z = 5 m; C = 0.8, nd = 0.02 and p = 100 mm/min, into cell DC 33 and DC 34 shall be inputted: z = 7 m, C = 0.3, nd = 0.02 and p = 100 mm/min, into cell DC 41 and cell DC 42 shall be inputted: z = 5 m; C = 0.8, nd = 0.02 and p = 100 mm/min, and into cell DC 44 shall be inputted: z = 7 m, C = 0.3, nd = 0.02 and p = 100 mm/min. Since a single value belongs to every item asked by the cell, each of these cells will directly render the input data as its attributes. There are, as may be obvious from Figure 2b, cells into which more than a value (a multi-value) shall be inputted for any items it may ask. In the present model, cells of this kind are: DC cell 12, 13, 22, 23, and 43. Each of these cells will average those multi-values as they are entered into it before rendering the result as its attributes. Table 1 displays the attribution of all these cells and final result of data preparation. Entries printed in bolds shown in the table are rendered by the respective cell as its attributes.

Table 1. Attribution of DCs objects and Final Result of Data Preparation

DC Initials	**z (m)**			**C**			**nd**	**p (mm/min)**
	Value 1	Value 2	**Average**	Value 1	Value 2	**Average**		
11	5.00		**5.00**	0.80		**0.80**	**0.02**	**100**
12	5.00	7.00	**6.00**	0.80	0.30	**0.55**	**0.02**	**100**
13	7.00	10.00	**8.50**	0.30	0.70	**0.50**	**0.02**	**100**
14	10.00		**10.00**	0.70		**0.70**	**0.02**	**100**
21	5.00		**5.00**	0.80		**0.80**	**0.02**	**100**
22	5.00	7.00	**6.00**	0.80	0.30	**0.55**	**0.02**	**100**
23	7.00	10.00	**8.50**	0.30	0.70	**0.50**	**0.02**	**100**
24	10.00		**10.00**	0.70		**0.70**	**0.02**	**100**
31	5.00		**5.00**	0.80		**0.80**	**0.02**	**100**
32	5.00	7.00	**6.00**	0.80	0.30	**0.55**	**0.02**	**100**
33	7.00		**7.00**	0.30		**0.30**	**0.02**	**100**
34	10.00		**10.00**	0.70		**0.70**	**0.02**	**100**
41	5.00		**5.00**	0.80		**0.80**	**0.02**	**100**
42	5.00		**5.00**	0.80		**0.80**	**0.02**	**100**
43	5.00	7.00	**6.00**	0.80	0.30	**0.55**	**0.02**	**100**
44	10.00		**10.00**	0.70		**0.70**	**0.02**	**100**

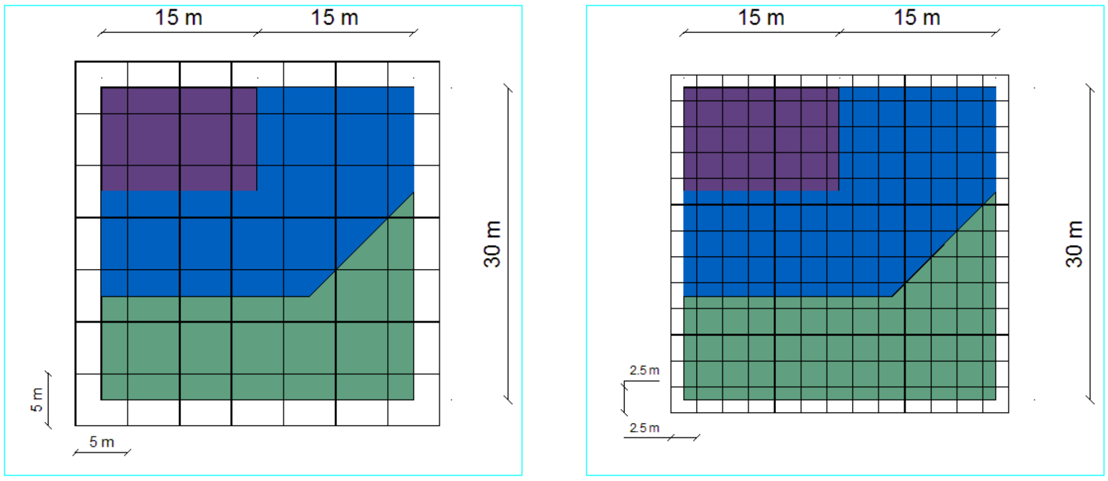
The subsequent step of the operation is the generation of Terrain_Cells by the object Terrain Cell Generator, in number equals to the number of DC cells already in existence, and is paired with each of them. Each Terrain_Cell adopts attributes from its DC partner as its own attributes. Until this stage, there exists in this module therefore 16 Terrain_Cells apart from the 16 DCs, whose attributes is in one to one correspondence to those of the DCs. It needs emphasize here that if so wished, every stage of the operation can be published by the Terrain Cell Publisher to give to the modeler information about the real-time development of the model he/she is constructing.

DISCUSSION

Careful readers may have noticed that there are several points, in connection with the construction of the present module, that need further considerations. Following points have been identified by the author as needing further consideration and will be brought into discussion in this section: (1). Edge Problem; (2). Faithfulness of the Model to Replicate the Terrain, (3). Representation of the Variation of Precipitation Intensity, and (4). Manual Data Entry. Let us begin with the edge problem.

Edge Problem

Edge problem is a set of problems in terrain modeling that is associated with the meshing of the terrain unto a network of a number of smaller elements. Two sub sets of this problem may be identified, namely (a) that which is associated with outer boundary of the terrain (peripheral edge problem) and (b) that which is associated with the boundary between two or more parts of the terrain that possess different features one to the other (internal edge problem). Cause of this problem lays in the fact that since elements employed in any terrain model are in regular shape (triangular, quadrilateral or rectangular (as in this module), etc), edge of the terrain can not be perfectly matched by the edge of the mesh. The same can be said of internal edge. Figure 2b above clearly shows these two sub-sets problems. It may be easily seen in the figure that due to shape regularity of the element employed in this model, outer boundary of the DCs network falls at a certain offset (5 m) from the actual terrain boundary, and since, for example, internal border between bituminous pavement and play ground falls on the middle of cells 12, 22, 32 and 43,



(a). Meshing the Terrain unto 49 Square Elements (b). Meshing the Terrain unto 169 Square Elements

Figure 3. Meshing the Terrain unto Higher Number of Cells

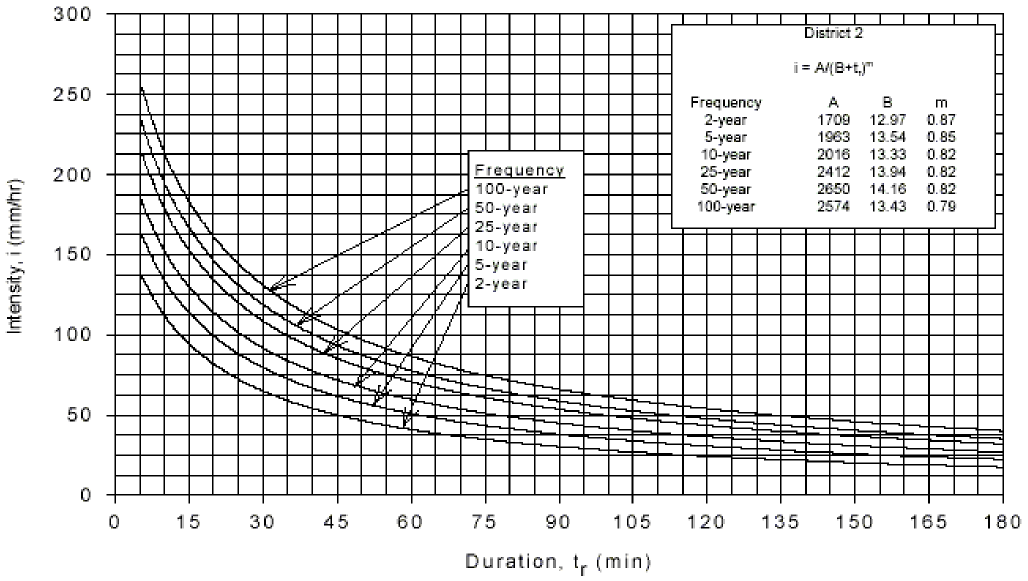
representation of their edge by the network of cell is bound to be imprecise. This problem can be overcome either by employing non-uniform shape cells (as for instance some are quadrilaterals while other triangles) in such a way as to match as close as possible the edge of the mesh unto that of the terrain, or by employing higher number of elements (hence smaller size of elements), or by employing both ways in combination. Figure 3 shows that by employing higher number of elements (hence smaller size elements) degree of impreciseness that arise from both peripheral and internal problems may somehow be reduced.

Faithfulness of the Model to Replicate the Terrain

Internal edge problems discussed in the preceding assumes another problem that relates to faithfulness of the replication of the terrain by the model. Referring to cells that contain internal borders, (cells of this category contains more than one type of terrain-shed), question may be asked as to which of the terrain-sheds shall the cell represent? It is of common knowledge that for every feature of a terrain-shed, a cell can only represent but one value. Should the cell represent only one of the terrain-shed, faithfulness has crept in, in regard to the unrepresented one. In this present module, faithfulness of replication is enhanced by taking average of the values of two or more terrain-sheds in a cell. This way of enhancing replication faithfulness is accepted only when areas of each terrain-shed contained by the cell are equal. At inequality, as is often the case, averaging does not faithfully represent the correct value. To represent correct value, taking weighted average of terrain-shed values in terms of contained area of terrain-shed within the cell may somehow enhance faithfulness of the replication, and can be implanted as a method in class: Data-Cell in place of the method of normal averaging. However lack of replication faithfulness is still bound to occur, despite employing the later method, if features of the terrain in question are of very high variation. As terrain to be modeled can be of very high variation, inventing methods which can precisely represent a terrain is mandatory. This study therefore strongly encourages engineers and scientists who are interested in urban terrain modeling, to do research for better way that can precisely and faithfully represent a terrain in an object-oriented model. Can fuzzy mathematic help to solve this problem? Finally, it should be mentioned that faithfulness of replication can also be boosted to a certain satisfying degree by employing smaller size of elements (hence greater number of elements).

Variation of Precipitation Intensity

It is commonly known that intensity of rain is not constant but varies over time during which it occurs, as may be shown by a bunch of intensity vs duration curves in Figure 4. In current module, rain intensity has been assumed as constant over time. In order to be more realistic, instead of a constant number, rain intensity should be represented by an equation, which is construed in reference to relevant Intensity vs Duration Curves, and be implanted into DCs as a method of representing precipitation features of a terrain.



District 2

$i = A/(B+t_r)^m$

Frequency	A	B	m
2-year	1709	12.97	0.87
5-year	1963	13.54	0.85
10-year	2016	13.33	0.82
25-year	2412	13.94	0.82
50-year	2650	14.16	0.82
100-year	2574	13.43	0.79

Figure 4. An Example of Rain Intensity vs Duration Curves (Akinyemi, 2006)

Manual Data Entry

As also may have been noticed by the readers, data entry in the operation of this module is still manually done. This certainly reduces the automations of an object-oriented computer modeling, and hence ways should be invented to increase the automation of data entry. In pursuant to it, this study encourages researchers to do research to come up with an automate ways by which a preparatory module of an object-oriented terrain modeling tool can capture physical features of a terrain directly from its satellite photograph or airborne electronic distance measurement. Optic Character Recognition (OCR) in combination with Artificial Neural Network (ANN) can be very useful for this purpose.

CONCLUSIONS

1. This research had developed an object-oriented preparatory module to use in conjunction with an object-oriented urban terrain hydrological response model;
2. Principle functions of the module are to capture an urban terrain's physical, hydrological as well as demographical data etc, and to arrange and prepare them as to become fit input to the core part of an urban terrain model;
3. The module comprises five classes of objects, each of which was predicated in such a way as to render the module capable to fulfill the purpose mentioned in point 2 above;
4. The module meshes the terrain into a network of a number of square cells, called Data-Cells (DCs), each of which is an object in its own right that can be predicated to match the property of part of the terrain that it represents and is capable of communicating with other objects and behaves autonomously;
5. Apart from Data-Cells, the module also generate Terrain_Cells (TCs) in numbers equals to the number of Data-Cells, and posses identical attributes as those of Data-Cells;
6. A Terrain_Cell mimics hydrological response and activities of part of a terrain that it represents, in such a way that an urban terrain with its associated hydrological activities will be faithfully represented in the form of a model when a definite system of Terrain_Cells work together for the terrain in question;
7. A Terrain_Cell is an object that connects this preparatory module to the core part of an urban terrain model by belonging to both the preparatory module as well as to the core part of the module. In such a way, it can be conceived of as a physical "port" or "a socket" through which the two parts of the model, the preparatory and the core, are connected to each other;
8. Edge problems, associated with the meshing of the terrain had been identified as reducing the faithfulness of replication. In response to it, utilizing higher number of cell network had been advised to overcome the problem;
9. Moreover, by employing greater number of cells (hence smaller size of cell) to represent the terrain in question, faithfulness of the replication of the terrain is greatly enhanced;
10. To further boost the faithfulness of data capturing, this research advised that method of weighted average in terms of different terrain-shed area within a cell, be developed and engaged into the module, in place of normal average it currently employs;
11. This module still assumes constant precipitation intensity throughout time, in spite of actual variation of the same over time; this may reduce the faithfulness of replication of the terrain by the module, in terms of rain intensity. To overcome this shortcoming, use of a rain intensity as function of time is imperative in place of a constant number, and
12. Finally, researchers are strongly encouraged to come up with a more automate ways of terrain data capturing, in place of the manual one still utilized in this current module.

References

Akinyemi, E. O. (2006). Urban Hydrology Lecture Notes. *Urban Hydrology* . Delft, The Netherlands: Unesco IHE.

Dia, H. (2001). An Object-Oriented Neural Network Approach to Short-Term Traffic Forcasting. *European Journal of Operation Research* , 253-261.

Mobley, A. V., Caroll, M. P., & Canann, S. A. (1998?). *An Object-Oriented Approach to Geometry Defeaturing for Finite Element Meshing.* Canonsburg PA: ANSYS Inc.

Pah, J. J. (2013). *An Object-Oriented Compter Software-Tool for Water Safety Plans.* Saarbrucken, Germany: LAP LAMBERT Academic Publishing.

Pah, J. J. (2011). Object-Oriented Approach for the Moment Curvature Relationship of a Varying Width and Multi Material Beam Cross Section. *Jurnal Teknik Sipil* , 18-31.

Poo, D., Kiong, D., & Ashok, S. (2008). *Object-Oriented Programming and Java* (2nd Edition ed.). Singapore: Springer Verlaag London Limited.

Struthers, A., & Illidge, J. (1995). Risk Assessment: An Object-Oriented Tool for the Rapid Evaluation of Liqud Spill Potential and Loss Prevention. *Computers Chemical Engineering* , 5453-5458.

Sydelco, P. J., Hlohowskyj, I., Majerus, K., Christiansen, J., & Dolph, J. (2001). An Object-Oriented Framework for Dynamic Ecosystem Modeling: Application for Integrated Risk Assessment. *The Science of the Total Environment* , 271-281.

Tucker, G. E., Lancaster, S. T., Gasparini, N. M., Bras, R. L., & Rybarczyk, S. M. (2001). An Object-Oriented Framework for Distributed Hydrolic and Geomorphic Modeling Using Triangulated Irregular Networks. *Computer and Geosciences* , 959-973.

Weidl, G., Madsen, A., & Israelson, S. (2005). Applications of Object-Oriented Bayesian Networks for Condition. *Computer and Chemical Engineerng* , 1996-2009.

Wong, Y. Y., & Yong, S. C. (1999). A Simulation Model for Container-Terminal Operation Analysis using an Object-Oriented Approach. *International Journal for Production Economics* , 221-230.

Appendix 1: General Structure of a Terrain_Cell Object

Terrain_Cell		
Attributes	Name: “mn”	
	Position_Coordinates “(x;y)”	
	Elevation “(z)”	
	Surface’ type (C)	C = Coeficient of Perviousness
	Cell_side’s_width “(s)”	
	Height of Perciptation “(p)”	
	Degree of Smoothness “(n_d)”	
Method1	Get_Time()	
	Clock.Instant_TimeInterval (t)	
Method2	Get_precipitation()	
	Precipitation.Height_of_Precipitation (h)	
Method3	Get_neighbours_elevation_and_Distance()	

Terrain_Cell	
	Neighbour_Identifier.neighbour_positions(Position_Coordinates “(x, y, z)”)
	Get_neighbours_coordinates {N=$(x, y, z)_N$; NE=$(x, y, z)_{NE}$; E=$(x, y, z)_E$; SE=$(x, y, z)_{SE}$; S=$(x, y, z)_S$; SW=$(x, y, z)_{SW}$; W=$(x, y, z)_W$; NW=$(x, y, z)_{NW}$}
	Calculate_neighbour_distances {N=r_N; NE=r_{NE}; E=r_E; SE=r_{SE}; S=r_S; SW=r_{SW}; W=r_W; NW=r_{NW};
Method4	Calculate_neighbours_slope()
	Neighbours_slopes {N=i_N; NE=i_{NE}; E=i_E; SE=i_{SE}; S=i_S; SW=i_{SW}; W=i_W; NW=i_{NW}}
Method5	Calculate_WaterSharingFactor()
	Numbers_of_RecievingNeighbours {R_n}
	WaterSharingFactor {$W_s=1/R_n$}
Method6	Calculate_IncomingWater()
	WaterVolume_of_Precipitation {$V_w=s^2h$}
	WaterVolume_from_ContributingNeighbours
Method7	Calculate_OutgoingWater()
	To_RecievingNeighbours {(N) = W_s f(i, r)$_N$; (NE) = W_s f(i, r)$_{NE}$; (E) = W_s f(i, r)$_E$; (SE) = W_s f(i, r)$_{SE}$; (S) = W_s f(i, r)$_S$; (SW) = W_s f(i, r)$_{SW}$; (W) = W_s f(i, r)$_W$; (NW) = W_s f(i, r)$_{NW}$;
	Infiltration { I = f(C,)}
Method8	Calculate_StayingWater()
	StayingWater = IncomingWater – Outgoing Water
	Height_of_StayingWater = StayingWater/s^2
Method9	StayingWater()
	Produce Height_of_StayingWater (H_{cell})
Method10	Publish_Height_of_StayingWater
	TerrainPlatform.Publish Height_of_StayingWater {(x, y, z, H_{cell})}