Dobesova Z.(2013): Using the " Physics " of Notation to Analyse ModelBuilder Diagrams.
SGEM 2013 13th International Multidisciplinary Scientific GeoConference,
Proceedings Volume I STEF92 Technology Ltd. Sofia, Bulgaria, 595 – 602s.
ISBN 978-954-91818-9-0, DOI: 10.5593/SGEM2013/BB2.V1/S08.039

USING THE "PHYSICS" OF NOTATION TO ANALYSE MODELBUILDER DIAGRAMS

Ing. Zdena Dobesova, Ph.D.

Palacký University, Olomouc, **Czech Republic**

ABSTRACT

Visual representations for data processing are used in geographic information systems. ArcGIS has components ModelBuilder for the creation of dataflow diagrams. ModelBuilder belongs to the visual programming languages. D. Moody proposed a set of principles for visual representation named "Physics" of Notation. Principles are based on wide review of relevant literature in cognitive psychology and software engineering. Moody presents nine principles. The applying of some principles to current notation in ModelBuilder for ArcGIS is presented in this article. Moody`s "Physics" of Notation framework is a way for discovering improvements in diagram notation for ModelBuilder. Besides them, this framework is an analysis that brings useful recommendations for creators of diagrams to increase the cognitive and perceptual quality of diagrams.

Keywords: geoinformatics, notation, visual programming language, diagram, cognition, perception,

INTRODUCTION

ArcGIS GIS product has the component for visual programming ModelBuilder. This component is aimed mainly for easy programming of the data batch processing by graphical way. Advantage is storing the program of batch processing for the repetitive running on various spatial data sets. The conversion of a visual program to a textual program can be used as introducing to the textual programing in Python for a novice programmer, in case of ArcGIS [1].

For all that purposes, it is necessary to use cognitively effective visual notations. Cognitively effective means optimized for processing by the human mind. That optimized notation help to user very quickly understand to the diagram.

Different methods for the evaluation of visual languages were published. Blackwell [2], [3] designed The Cognitive Dimensions Questionnaire with open questions. The Daniel Moody theory is newer theory especially for graphical languages and arose from scientific background.

The Mood's theory of Physics of Notation was synthesized from theory and empirical evidence drawn from a wide range of fields, including communication, semiotics, graphic design, visual perception, psychophysics, cognitive psychology, education, HCI, linguistics, information systems, cartography and diagrammatic reasoning [4]. Physics of Notations, as it focuses on the physical (perceptual) properties of notations rather than their logical (semantic) properties. Semantic properties of symbol in ArcGIS ModelBuilder are described in software help and in several articles [5], [6], [7].

Example of diagram from ModelBuilder is in fig. 1.

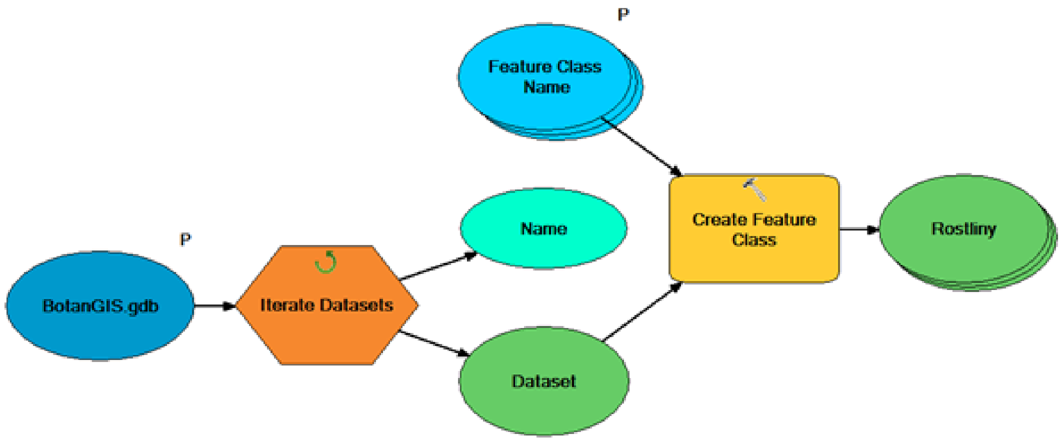

Fig. 1 Diagram (model) with iterator to create feature classes in dataset

PRINCIPLES OF PHYSICS OF NOTATION

Moody presents nine principles that can be successfully used to evaluate visual programming languages. It is also way how to compare, and improve existing visual notations. Principles are arranged as the diagram with nine hexagons (Fig. 2). The modular structure of physic of notation is designed to make it easy to add or remove principles, emphasizing that they are not fixed or immutable but can be modified or extended by future research [8].

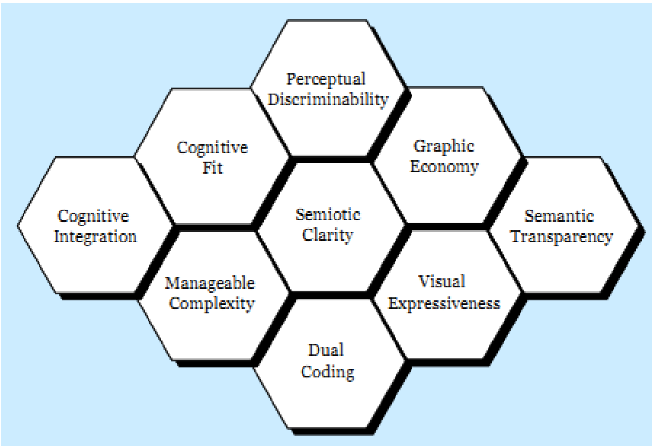

Fig. 2 The Physics of Notations: Principles for Designing Cognitively Effective Visual Notations [8]

The article takes some principles and applies it to the graphic elements of ModelBuilder to evaluate it.

Principle of Semiotic Clarity

The first and central principle is the principle of Semiotic Clarity. The principle of Semiotic Clarity defines that there should be a one to one correspondence between syntactic and semantic features (Fig. 3). According to that principle it is not allowable a symbol redundancy, a symbol overload, a symbol deficit or a symbol excess.

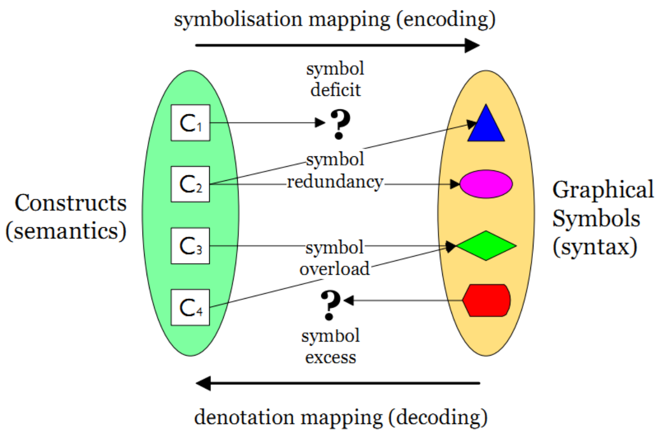

Fig. 3 Principle 1: Principle of Semiotic Clarity [4]

ModelBuilder used the oval symbol for data, rectangles for process (tool) and hexagon for iterator. The symbols vary in shapes (oval, rectangle, hexagon). The fill colors also differ (Fig. 4). There semiotic clarity is at very good level in ModelBuilder.

There is the possibility to change the shape of symbol to another shape. The graphical editor offers to change the default shape to the star, diamond, circle, rectangle etc. Some users try to change some symbol to another shape to express e.g. important input data. The change causes the symbol redundancy and the reader of the diagram is confused. The recommendation for a user is to keep the default symbol in ModelBuilder. Some "creative work" destroys the original setting.

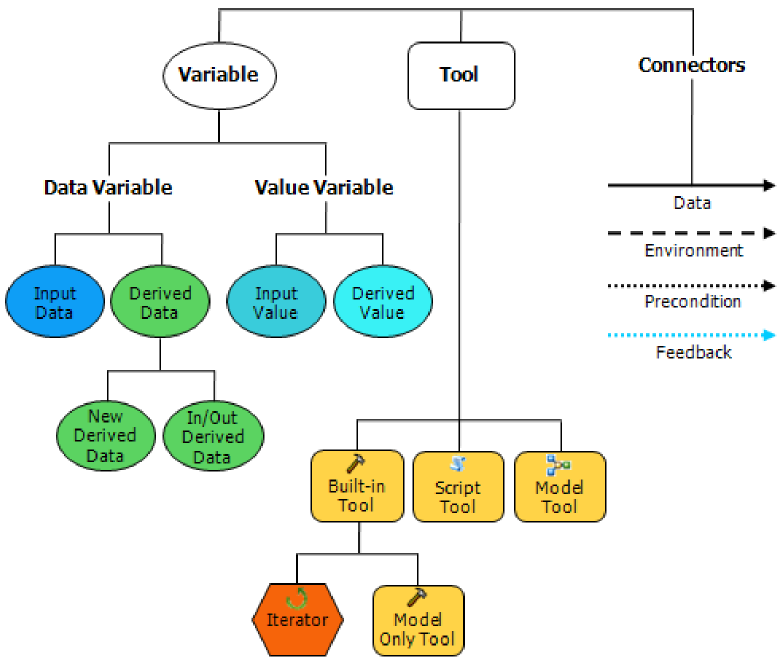

Fig. 4 Graphical symbols in ModelBuilder

Principle of semiotic clarity strongly interact with the next two principles, Perceptual Discriminability and Semantic Transparency [9].

Principle of Perceptual Discriminability

The principle of Perceptual Discriminability states that different symbols should be clearly distinguishable from each other [10]. The symbols differ in ModelBuilder by shape and color. Rectangle for the process has only yellow color. The data has green, light blue or dark blue fill of ovals. The green and blue colors are optical near. Orange color is used only for iterator. The combination of color and shape increase the discriminability.

Moreover, the inner icon is used in shapes. Orange hexagon of iterator contains an icon with small circular arrow. That circular arrow expresses the repetition of operation (cycles). Yellow rectangle contains small icon of hammer to emphases the meaning. The rectangle expresses function (tool) or process. Besides the hammer, other icons are used. Icon of paper to expresses script in a programming language. A small icon of model expresses a nested model. All yellow boxes mean processing data. Small icon distinguishes the type of process.

The comparing every graphic element against all other graphic elements decides if symbols are perceptually discriminable. The graphic elements are shown pairwise in the table 1 to make judging easier. The question is: Are the symbols are easily distinguishable?

The discriminability of yellow boxes (tools) is low only by small inner icon. Also, discriminability of input data, new derived data, In/Out data, input value and derived value is low. Only fill color discriminates them.

Table 1 Pairwise comparing of graphic elements

Graphical Element 1	**Graphical Element 2**	**Perceptual Discriminability**
Input Data	Built-in Tool	Yes
Iterator	Input Data	Yes
Iterator	Built-in Tool	Yes
Input Data	New Derived Data	No
Built-in Tool	Script Tool	No
Script Tool	Model Tool	No
Derived Value	Input Value	No

Very good discriminability is when input data (or value) are parameters of a model that means that model is not hard codded, only for definite data. It is expressed by “P” letter at right upper position near oval. The experiment with eye-tracking equipment was processed to verify some expectation in the evaluation of properties of graphic notation for ModelBuilder. An analysis of eye movements (eye-tracking) is one of the most objective methods[11]. The fig. 5 shows the heat map. Testers were asked to find the parametric input data. The letter “P” is in the middle of the hot spot.

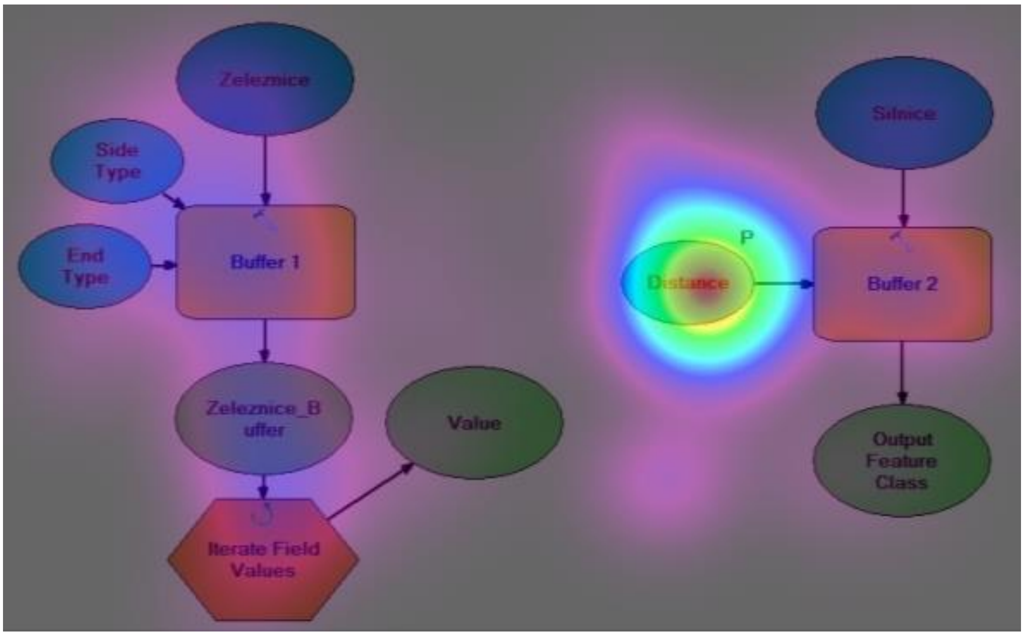

Fig. 5 Heat map for the model with parametric input data - P

Principle of Semantic Transparency

Principle means - use visual representations whose appearance suggests their meaning. The principle of Semantic Transparency demands that it should be possible to infer the meaning of a symbol from its appearance. The benefit is “reduced cognitive load because they have built-in mnemonics", which leads to improved speed and accuracy of understanding the information contained in a diagram. The use of iconic representations additionally improves the speed of recognition and recall.

Processes (tools) are expressed by rectangles and iterators are expressed by hexagon in ModelBuilder. Both these graphical elements are used in standard flow charts for expressing commands and cycles with condition. The reader of the diagram is familiar with these symbols.

One new idea is: Put to the data symbol an additional icon that expresses the type of data (Fig. 5). The basic division of data is into vector data and raster data in GIS. Small inner icon in oval can express the type of data (vector and raster data). These small icons are used in ArcCatalog. User knows these icons and understands to their meaning. This idea increases the semantic transparency. There is a space to suggest whole list of icons for point, line, polygon data, annotation, relation etc.

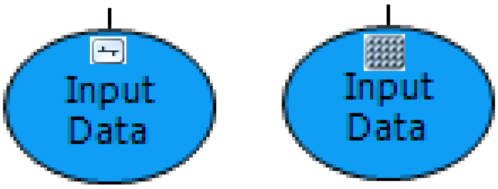

Fig. 6 Additional inner icons express the type of data (vector line data-left, raster data – right)

Principle of Dual Coding

Principle recommended - use text to complement graphics. Textual labels are used in graphic elements. The name of tools (Buffer, Clip, Create Feature Class) are introduced automatically in the design stage. The text labels help to user to understand the diagram. Labeling can be considered as dual codding. The label can be changed from the default name. The change of name for the tool is not recommended. It decreases the understanding. The principle of dual codding is used in the right way in ModelBuilder.

Also, the input/output data are automatically labeled according to the name of data (river, road, wood etc). When the diagram is for various data it is better change labels to universal label than using concrete name of a layer. It is also necessary to consider the length of text labels. Very long labels require bigger graphical elements. The size disproportion of graphical element in one diagram decrease aesthetic and perceptual quality.

The specific color fill for specific graphic symbol can be assumed also as dual coding. The combination of shape and fill color increase the visual distance between symbols. Redundant coding is an important technique in communication theory to reduce errors and noise.

Principle of Visual Expressiveness

Principle of Visual Expressiveness states that the full range of visual variables and their full capacity should be used to represent the notational elements. The evaluation according to this principle can be by only Yes/No result for each variable.

The color is one of the most effective visual variables. The color is used as fill of graphic elements. The different colors are used for different symbols. Next visual variable, shape is also used in ModelBuilder. Base shapes of graphical element are box, oval and hexagon.

Another visual variable is the size. The size of the symbol is changeable by user opinion, especially when the label does not fit to the oval. A big increasing of size symbol can be assumed as expressing of bigger meaning [13]. In fact, a signification of the data is the same in diagram.

The visual variable orientation is used only on connection arrows. Arrows express the flow of data. The variable orientation is not used for graphic elements (box, oval, hexagon). Last two variables are texture and brightness. They are not used. The brightness of graphical symbols is medium and near the same for all the elements.

Principal of Graphic Economy

That principle detects if the number of different graphical symbols should be cognitively manageable. The symbol choice affects the ease of memorizing and recalling visual diagrams [12]. The number of base graphical elements is three and connectors in ModelBuilder. Moreover, there are some modifications of base graphic elements (various fill color for data and variables, inner icons). The introducing of inner icon for data in principle of semantic transparency increases the number of graphical symbols. The number of the symbol is well cognitively manageable in ModelBuilder.

Principle of Cognitive Integration

This principle helps to answer to the question as: Where am I in diagram? It is possible switch on small overview window in ModelBuilder. Overview window helps in navigation.

There is no possibility to change the level of detail in diagram (drill-down, roll-up). Only text labels could title parts of the diagram to optical separate different part.

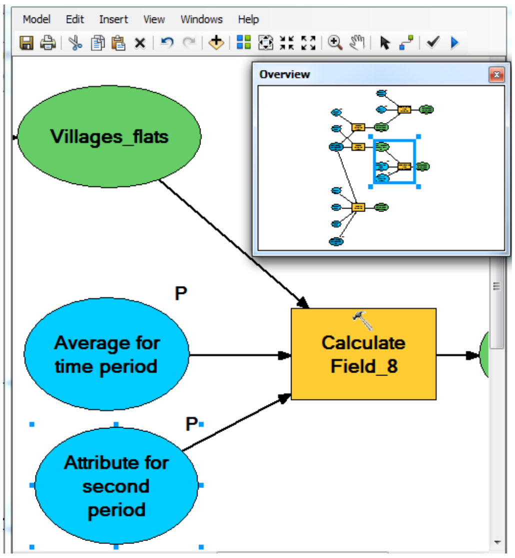

Fig. 6 Detail part of the model and the overview window with whole model

Last two principles Complexity Management, Cognitive Fit did not apply.

CONCLUSION

Daniel Moody postulated nine principles that aid in the development of proper graphical syntax. These nine principles form a modular framework for the development and assessment of a cognitive effective visual language. Seven principles were applied to diagrams in ModelBuilder. Semiotic Clarity is good in ModelBuilder. Perceptual Discriminability is higher between symbols for tools and data than between symbols for input output data. ModelBuilder offers the change of graphical symbol. The mixture of shapes for the same construct introduces symbol redundancy and would decrease the aesthetic quality of the diagram [13].

The proposal of adding inner icons to data graphic symbol would increase the Semantic Transparency. Combination of different shape and fill color for graphic elements assures the principle of Dual Coding. From the point of Visual Expressiveness, the visual variables color, shape and orientation are used. The number of the symbol is well cognitively manageable in ModelBuilder. In the principle of Cognitive Integration, there is space for improvement. Finally, overall assessment of ModelBuilder is well from the point of principles for designing cognitively effective visual notations.

ACKNOWLEDGEMENT

Work was supported by the project CZ.1.07/2.3.00/20.0166.

REFERENCES

1. Dobesova, Z., Visual programming for novice programmers in geoinformatics, in 12th International Multidisciplinary Scientific GeoConference. 2012: Bulgaria, Albena. p. 433- 440.
2. Blackwell, A.F., Cognitive dimensions of notations: Understanding the ergonomics of diagram use. 2008: Herrsching. p. 5-8.
3. Green, T.R.G. and M. Petre, Usability analysis of visual programming environments: A 'cognitive dimensions' framework. Journal of Visual Languages and Computing, 1996. 7(2): p. 131-174.
4. Moody, D.L., The "Physics" of Notations: A scientific Approach to Designing Visual Notations in Software Engineering, in Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 2. 2010, ACM: Cape Town, South Africa. p. 485-486.
5. Dobesova, Z. Visual programming language in geographic information systems. In Recent Researches in Applied Informatics, 2nd International Conference on Applied Informatics and Computing Theory, AICT 11. 2011. Prague: NAUN/IEEE, WSEAS Press.
6. Dobesova, Z. Programming language Python for data processing. In Electrical and Control Engineering (ICECE), 2011 International Conference. 2011.
7. Dobesova, Z. and P. Dobes. Comparison of visual languages in Geographic Information Systems. In IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC). 2012.
8. Moody, D. Theory development in visual language research: Beyond the cognitive dimensions of notations. In Visual Languages and Human-Centric Computing, 2009. VL/HCC 2009. IEEE Symposium 2009.
9. Thomas, J.C., et al. Using the Physics of notations to analyze a visual representation of business decision modeling. In Visual Languages and Human-Centric Computing (VL/HCC), 2012 IEEE Symposium 2012.
10. Moody, D.L., The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. IEEE Transactions on Software Engineering, 2009. 35(6): p. 756-779.
11. Popelka, S. and V. Voženílek, Specifying of requirements for spatio-temporal data in map by eye-tracking and space-time-cube, in International Conference on Graphic and Image Processing (ICGIP 2012). 2013: Singapore p. 87684N-87684N.
12. Figl, K. Symbol choice and memory of visual models. In Visual Languages and Human-Centric Computing (VL/HCC), 2012 IEEE Symposium 2012.
13. Dobesova Z. Strengths and weaknesses in data flow diagrams in GIS, Diagram Aesthetics and Layout (Special Issue), Journal of Visual Languages and Computing (in review)