

©Copyright JASSS



[Daniel Kornhauser, Uri Wilensky and William Rand \(2009\)](#)

Design Guidelines for Agent Based Model Visualization

Journal of Artificial Societies and Social Simulation vol. 12, no. 2 1

[<http://jasss.soc.surrey.ac.uk/12/2/1.html>](http://jasss.soc.surrey.ac.uk/12/2/1.html)

For information about citing this article, click [here](#)

Received: 14-Aug-2008 Accepted: 01-Feb-2009 Published: 31-Mar-2009



Abstract

In the field of agent-based modeling (ABM), visualizations play an important role in identifying, communicating and understanding important behavior of the modeled phenomenon. However, many modelers tend to create ineffective visualizations of Agent Based Models (ABM) due to lack of experience with visual design. This paper provides ABM visualization design guidelines in order to improve visual design with ABM toolkits. These guidelines will assist the modeler in creating clear and understandable ABM visualizations. We begin by introducing a non-hierarchical categorization of ABM visualizations. This categorization serves as a starting point in the creation of an ABM visualization. We go on to present well-known design techniques in the context of ABM visualization. These techniques are based on Gestalt psychology, semiology of graphics, and scientific visualization. They improve the visualization design by facilitating specific tasks, and providing a common language to critique visualizations through the use of visual variables. Subsequently, we discuss the application of these design techniques to simplify, emphasize and explain an ABM visualization. Finally, we illustrate these guidelines using a simple redesign of a NetLogo ABM visualization. These guidelines can be used to inform the development of design tools that assist users in the creation of ABM visualizations.

Keywords:

Visualization, Design, Graphics, Guidelines, Communication, Agent-Based Modeling

Introduction

1.1

A major purpose of agent-based models is to visually convey the behavior of the model clearly and quickly. An effective visualization of an ABM identifies the important model elements and helps the user to understand the model's behaviors. However, designing an effective visualization can be challenging for model authors, most of whom do not have visual design training. Establishing principles for designing cognitively efficient, aesthetic and communicative visualizations can guide

researchers in their graphic design process of agent-based models. Cognitive efficiency ([Cobb 1997](#)) takes place when swift perceptual inferences replace slow cognitive comparisons ([Healey 1995](#)). A cognitively efficient visualization can facilitate the detection of important features, such as the discovery of emergent behavior, by providing an optimal representation to facilitate tasks such as cluster detection. Aesthetics is another key consideration for ABM visualization design. An aesthetic visualization can add visceral appeal ([Norman 2004](#); [Brath 2005](#)), thus improving the visualization by highlighting its message and increasing its memorability.

1.2

Some ABM visualizations are more easily understood than others and some are more aesthetic than others (see Figure 1). Developing effective guidelines to increase the accuracy and readability of the ABM visualizations can assist in the comprehension and appeal of the underlying model.

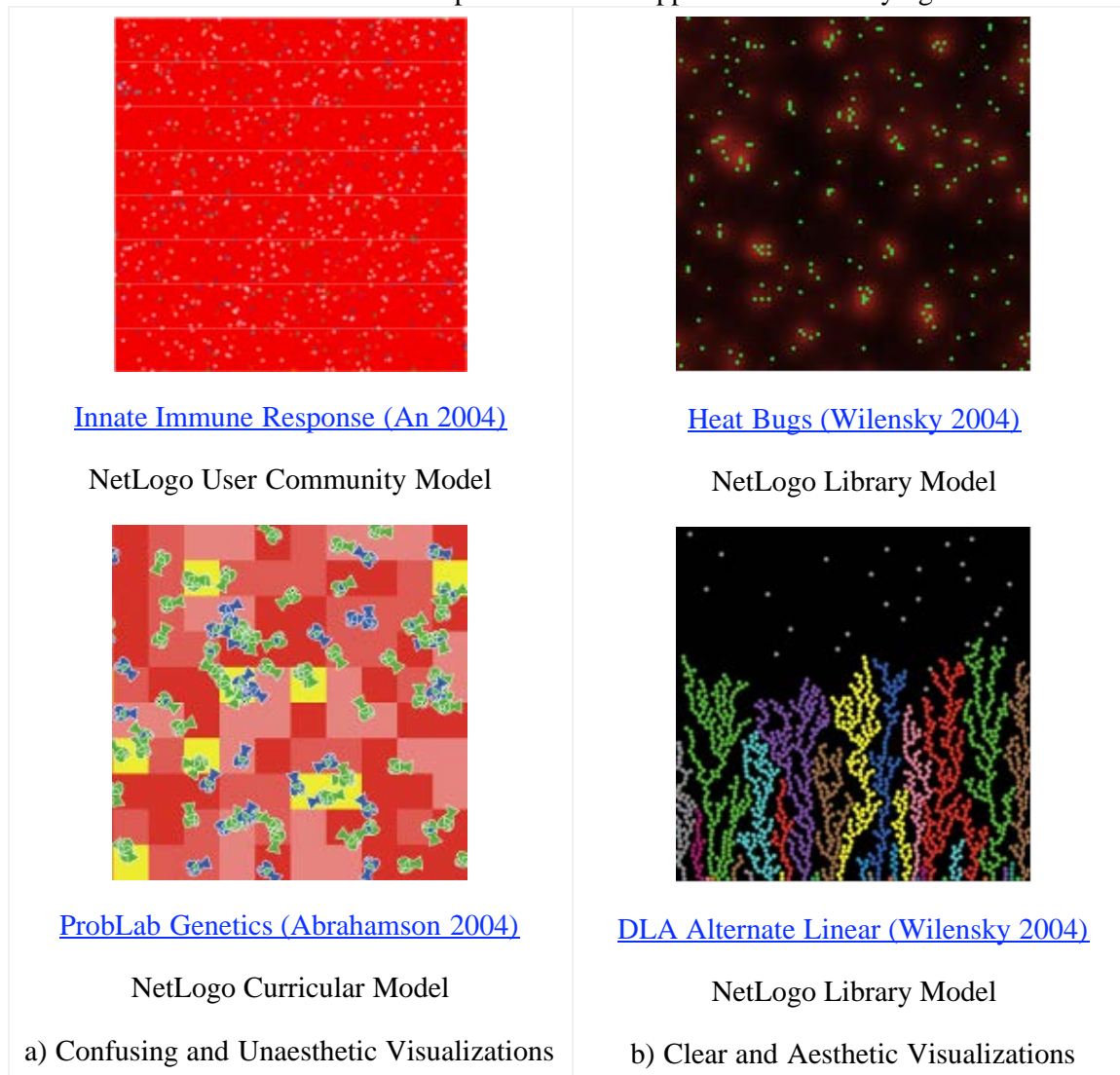


Figure 1. Examples of ineffective unaesthetic visualizations versus effective aesthetic visualizations in NetLogo. Note that in the visualizations on the left a), the authors chose to prioritize other criteria: In the Innate Immune Response Model, the author used a red garish background to represent the color of blood. In the ProLab Genetics Model, the author used neon colors given that they tend to captivate the interest of children. In this paper we prioritize cognitively efficient representations above accurate figurative representations or motivational representations.

1.3

The difficulty in creating a cognitively efficient and aesthetic ABM visualization resides in the open-ended and unconventional representation of agent based models. Other software used for modeling, such as Geographic Information Systems (GIS) or spreadsheets provide more canonical and

constrained representations such as maps in GIS or plots in spreadsheets. The design of these constrained representations is possible to automate, for example the computer can assist the user in creating maps and a wide variety of diagrams and plots. However, this is difficult to achieve in ABM visualization given the diversity of ABM depictions. The nature of agent-based modeling frameworks demands a wide array of graphic representations (see Figure 2) that are usually custom-made for each model. Thus, well-designed ABM visualizations requires an in-depth knowledge of visualization guidelines and the skill to use those guidelines to craft clear and aesthetic visualizations.

1.4

In this paper, we first describe and categorize ABM visualizations created in NetLogo ([Wilensky 1999a](#)). This categorization aims to introduce the reader to the diversity of visualizations in ABM and some of the better practices in ABM visualization design. Afterwards, we discuss how Gestalt psychology, semiology of graphics and scientific visualization can inform the design of ABM visualizations. Through this discussion we hope to introduce ABM developers to techniques for improving their visualizations such as, foreground/background segregation, informed use of visual variables, and removal of visual interferences. Then, we present the goals for applying these techniques: to simplify, emphasize and explain the visualization. Finally, we examine a redesign of an ABM visualization to illustrate these ideas.



Background

2.1

Visualization guidelines for an agent-based model draw from both aesthetic and cognitive design traditions. Aesthetic design techniques have a long history originating in artistic and graphic design developed over centuries. However, cognitive design techniques were only studied recently, originating with Gestalt psychology research in the early twentieth century.

2.2

Aesthetic visualization techniques involve color theory, image composition, and other visual considerations seeking to remove discordant structures and highlight a message. A detailed discussion of graphic aesthetics is outside the scope of this paper, but the reader can gain some basic knowledge in this area by consulting introductory references ([Pipes 2004](#); [Maeda 1999](#)). However, to effectively employ aesthetic design, reading art and design theory is insufficient, it is a craft primarily acquired by actively practicing with an expert. Nevertheless, aesthetic designs do follow certain general rules such as, balance, unity, harmony, scale and proportion, and contrast and emphasis. By understanding these design criteria the developer of ABM visualizations should be capable of articulating when, how and why a visualization is pleasing to the eye.

2.3

Cognitive visualization techniques tend to be more explicit than aesthetic techniques and therefore can be more easily conveyed and systematized. Cognitive visualization techniques have been developed in several fields, starting in Gestalt psychology ([Wertheimer 1923](#)).

2.4

Gestalt psychology is a theory of mind with the motto 'The whole is greater than the sum of its parts'. Gestalt psychologists studied with mechanisms of visual recognition of figures and whole forms from a collection of simple lines and curves.

2.5

Jacques Bertin developed another significant visualization framework in his seminal book, *The Semiology of Graphics* ([Bertin 1967](#)). In his work, Bertin, defines a coherent and comprehensive symbol scheme where he presents visual variables and studies the type and quantity of information they can convey.

2.6

There are many other significant frameworks that have been developed to study visualization, particularly in the field of statistics. ([Tukey 1977](#); [Tufte 1983](#)).

2.7

In the 1980s, with the advent of personal computers, scientific visualization became a distinct field,

building upon research in fields such as cognitive psychology and psychophysics to improve visualization techniques. Currently, a wide array of material is available, from comprehensive visualization textbooks ([MacEachren 1995](#); [Ware 2002](#)), to specialized papers that address specific topics such as real-time multivariate data visualization ([Healey, 1996](#)) or color schemes in computer displays ([Harrower & Brewer 2003](#)).

2.8

Capturing which of these techniques apply to ABM visualization design is difficult due to the multidisciplinary nature and breadth of visualizations. Additionally, prioritizing which techniques are more relevant than others increases the challenge. One way to solve this problem is to approach it in the same way that other disciplines such as art, design or architecture have done. These disciplines have advanced by documenting their best practices and by creating guidelines for their craft. In a similar vein, we will begin by studying and classifying visualizations created with NetLogo.

2.9

We use models from the NetLogo models library ([Wilensky 1999b](#)) as data for our classification. We chose to use the NetLogo modeling environment since it provides a wealth of ABM visualizations in its extensive models library and has a very large and active user community which also contributes and publishes models ([Wilensky & Reisman, 2006](#)). Our need for ABM visualization guidelines stemmed in part out of the desire to improve NetLogo's visualizations and to provide the NetLogo user community with visualization guidelines for submitting their models. Furthermore, NetLogo provides several key features that facilitate ABM visualization design. NetLogo is designed to be both 'low threshold' and 'high ceiling' ([Wilensky, 2001](#); [Tisue & Wilensky, 2004](#)). Consequently, NetLogo provides high-level primitives for programming and visualizing an ABM model. These high level ABM primitives liberate the programmer from implementing low level programming of control and data structures to address agents; giving more time to attend to the visualization design. Additionally, high-level graphics primitives within NetLogo enable users to rapidly prototype different visualizations in tight iterative cycles without having to learn the intricacies of low-level graphics libraries.



Guidelines

3.1

We present three sets of guidelines for ABM visualization design.

3.2

The first set of guidelines is an ad hoc classification of ABM visualizations currently found in the NetLogo Models Library. This classification can familiarize the novice with the breadth and bounds of existing ABM visualizations. This overview of visualizations can assist the modeler in starting to author his/her own model inspired by visualizations similar to the one she/he wishes to create. Often one of the easiest ways to design a visualization is to look at other visualizations that have similar properties and to emulate them in ways that make sense within the context of the new model.

3.3

The second set of guidelines presents well-known visualization principles in the context of ABM visualizations. By learning about these principles, model authors can remove or replace the elements of the design that cause confusion. These principles provide the authors with a firm grounding in how they can increase the visualization's comprehension.

3.4

The third set of guidelines sets clear objectives for applying the visualization principles. The modeler can improve the visualization by simplifying and emphasizing its message. These goals should always be kept in mind to ensure that the design principles are applied appropriately.



Non-hierarchical Classification of Agent Based Model Visualizations

4.1

There are now several ABM toolkits available. It is impractical to categorize all of the ABM visualizations of every model in each ABM toolkit, and therefore we have chosen to examine the NetLogo models library. Based on this extensive library (over 200 models) the authors have developed a set of distinct tags (see Figure 2-a 2-b 2-c). Our approach was inspired by ([Lohse, 1994](#)) who classified static visualizations but we did not use external subjects to perform the classification. We clustered screenshots of similar visualizations displayed in the NetLogo Samples Model Library. However, on some occasions if the screenshot did not clearly demarcate a category, we ran the model in NetLogo to observe the model's behavior. Due to the open-ended design of ABM visualizations we find it is impractical to try to classify them using a rigid taxonomy, therefore we created an ad-hoc non-hierarchical classification. The sole purpose of these classification tags is to orient the novice to the possibilities and limitations of ABM visualizations.

4.2

Labeling the NetLogo Models Library has several advantages. First, the visualizations are intrinsically driven by agents since NetLogo models are constructed with data structures and primitives implicitly imbued with ABM semantics. Moreover, NetLogo visualizations share the same graphical and programmatic primitives; thus they can be compared and analyzed using the same criteria. Finally, the NetLogo Models Library goes through a strict checkout process assuring that the models have sufficient quality and accuracy to be deployed for use in educational institutions.

4.3

This classification could be improved substantially, particularly by asking external subjects to tag the images and by using dynamic NetLogo animations instead of static images from the screenshots. However, we consider this first classification suitable for our purposes, namely, assisting the novice in locating a visualization category to initially guide his or her model's visualization design.

4.4

The classification of models is purely visual. It does not take into account the particular phenomenon being explored by the model. In other words, we did not consider the subject area or discipline of the model. Of course, some visualizations in the same category share the same type of phenomenon, but it is not unusual to find unrelated phenomena appearing in the same category. This classification is not meant to be a rigid comprehensive classification; on the contrary, it is meant to be an easily understandable and flexible overview. Thus, the modeler should envision the model that they eventually want to develop and then examine categories below based on that conception, to find a starting point for the design of their new visualization.

4.5

We divide ABM visualizations into three main categories: conventional, unstructured and structured. Conventional visualizations make use of previously established representations while unstructured visualizations display abstract or irregular patterns, and structured visualizations convey meaning through shape or regular patterns. These categories should not be conceived of as mutually exclusive. Indeed, visualizations can encompass several categories and can even switch categories depending on the state of the simulation. Many tags can describe one model, and the same tag can be used to describe many models. We created categories that convey a certain sense of the potential of ABM graphics to the novice.

4.6

Conventional ABM visualizations encompass established representations of mathematical and/or natural phenomena. The viewer understands these representations since he has seen or studied similar representations previously. This exposure may have taken place in several contexts such as an educational institution or popular culture. For instance, in recent years fractals have entered educational institutions and mainstream culture -- thus an educated audience can now understand the context and relevance of a fractal in a visualization when previously they would have been unable to comprehend the representation.

4.7

We divided the rest of ABM visualizations into two categories: structured ABM visualizations and unstructured ABM visualizations. Note that this is not a rigid categorization; an unstructured ABM visualization can become a structured ABM visualization through emergent behavior. These

visualizations are not understood due to a predetermined convention; the viewer uses his visual intuition and the documentation accompanying the visualization to process and give meaning to the visual features. The aim of our classification is to differentiate the key mechanisms the viewer uses for understanding a visualization. In structured visualizations the shape or clusters formed by the agents plays a primal role, while in an unstructured visualization no contours or areas are markedly defined, instead the viewer uses the spatial distribution of the agents and their color, shape and orientation to make sense of the image.

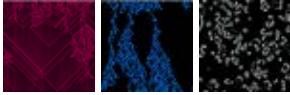
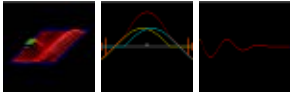
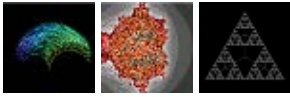

Tag Name	Visualization Examples	Short Description
Cellular Automata		Cellular automata consist of regular grids of agents where a finite number of states are usually mapped to a color.
Physics representations		Conventional physics representations have been historically used to illustrate natural phenomena.
Fractal		Fractals and L-systems have particular but well- established visualization techniques.
Geometrical		Geometrical patterns are generated through spatial interaction of agents.

Figure 2. Conventional ABM visualizations encompass well-established mathematical representations or natural phenomena. The viewer understands these graphical representations only because he has seen or studied similar representations previously

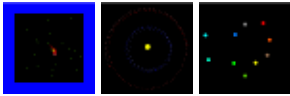
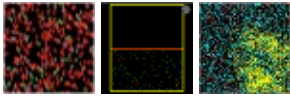



Tag Name	Visualization Examples	Short Description
Few interacting mobile agents (Turtles in NetLogo)		A few mobile agents driven by simple rules can display complex behavior.
Many interacting mobile agents (Turtles in NetLogo)		Many mobile agents can display complex spatial or color patterns.
Many interacting mobile and immobile agents (Patches and Turtles in NetLogo)		Many mobile agents in the foreground interacting with many immobile agents in the background. These visualizations can become structured visualizations after an emergent pattern arises.

Figure 3. Unstructured ABM visualizations appear as irregular spatial patterns. They mostly convey information through the perception of the change of color, texture or spatial distribution of the composition. They are mostly characterized by an irregular spatial distribution of agents.

Tag Name	Visualization Examples	Short Description
Flow simulations		Flow simulations are composed of agents that are perceived to flow (such as traffic simulations).
Aggregation		Aggregation is a phenomenon found in



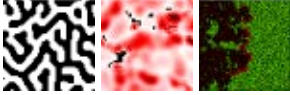
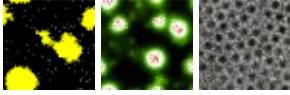
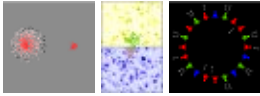


		biology or physics. We classify these models as regular structures formed by agents.
Emergent patterns with mobile agents (Turtles in NetLogo)		These systems start with interspersed random agents that over time create an emergent pattern. There are many types of emergent patterns such as clustering or types of synchronization.
Emergent patterns with immovable agents (Patches in NetLogo)		These models are subsets of cellular automata, but they often model concrete natural phenomena instead of abstract mathematic phenomena.
Emergent Pattern with Patches and Turtles		In these models the interaction between mobile and immobile agents creates clusters. Note that these can become unstructured visualizations if the clusters disperse.
Figurative		Figurative models represent real world objects that are not merely clusters of agents: a cell, a plant, and a dinner.
Particle Systems		In particle systems, each agent is driven by the interaction of its physical properties such as mass and shape with external forces such as gravity, wind or viscosity.
Abstract		These visualizations do not represent any natural, physical, or mathematical phenomenon but use a canonical representation.

Figure 4. Structured ABM visualizations form an abstract or figurative shape or regular pattern. These visualizations are characterized by a regular spatial positioning of agents creating clusters, regions, aggregations, or particle trajectories.

Design Principles

5.1

After completing an initial design, the iterative redesign of an ABM visualization can be divided into two parts: removing confusion and adding clarity. Removing confusion can be achieved by eliminating cognitive and aesthetic obstacles. Adding clarity can be accomplished by incorporating or reinforcing visual cues to emphasize the model's key variables and by increasing its aesthetic appeal. These enhancements should result in an unambiguous, memorable and pleasing visualization, which allows the viewer to easily focus on the main message of the model.

5.2

At a high level, cognitive hurdles can arise from an unintended holistic perception of agents. This often occurs when the author's visualization fails to take into account Gestalt principles. For example, the use of figurative icons instead of abstract shapes will impede the viewer's perception of the image as a whole. You can observe, in Figure 5-a, that circles tend to merge together to produce new shapes, while the human icons overlap, but do not merge. Gestalt psychologists observed this phenomenon when they first studied visualizations. They observed that it is easier to perceive a whole given the

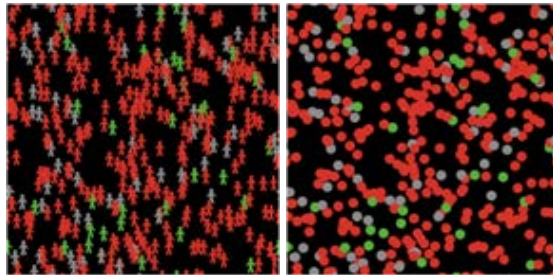
merger of abstract shapes rather than it is to perceive a whole from set figurative objects that perceptually overlap but do not merge.

5.3

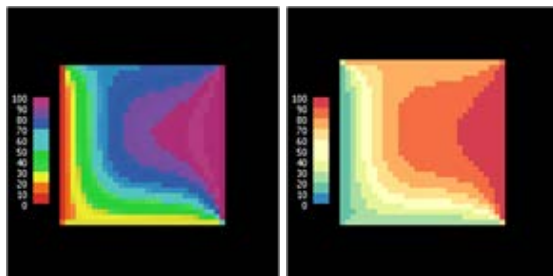
At a lower level, semiotic hurdles can occur as misrepresentations due to erroneous mappings from model variables to their visual portrayal. For example, the use of an unintuitive color scheme can confuse the viewer by mapping a numeric value to an unexpected color (A color scheme maps values such as: 10, 20, 30, 40, to colors such as: red, yellow, green, blue). This often happens in toolkits that do not provide pre-constructed color schemes since the modeler is left on its own to assemble color schemes with color systems such as RGB or HSV. However, building a color scheme requires deep knowledge about color representations hence is difficult to do without prior experience. For example, the construction of a divergent spectral or "rainbow" color can be a challenge as seen in Figure 5-b. Many requirements have to be satisfied such as placing the yellow color in the middle of the scheme, avoiding the confusion of the outermost purple and red colors, and being careful to map the red color to the highest numerical value even if it has the lowest numerical hue. This issue is addressed in many environments such as Geographic Information Systems (GIS) platforms (e.g., ArcView), or Visualization Toolkits (e.g., VTK), by providing pre-built color schemes assembled around a perceptual correspondence. ColorBrewer is an online tool designed to help people select perceptually tested color schemes ([Harrower & Brewer 2003](#)). Seen in Figure 5-c, ColorBrewer is an excellent starting point for color scheme selection. Thus, scientific software such as R, MATLAB or Mathematica have adopted ColorBrewer schemes. We also created a NetLogo extension to facilitate the selection and mapping of ColorBrewer schemes in NetLogo (<http://ccl.northwestern.edu/papers/ABMVisualizationGuidelines/palette/>)

5.4

Finally, at the lowest level, perceptual impediments arise when we exceed the limitations of our low-level visual system. Visual features that are difficult to distinguish can disable our pre-attentive processing capabilities. Pre-attentive processing can be hindered by other cognitive phenomena such as interference between visual features ([Healey 2006](#)), or temporal aliasing ([Bach 2006](#)) (see Figure 5-d).



a) The human silhouette icons on the left overlap, while the circle shapes on the right merge to create new compound shapes. Thus, it seems easier to perceive new compound shapes on the right than in the left. This Virus model ([Wilensky 1998a](#)) is at <http://ccl.northwestern.edu/netlogo/models/Virus>.

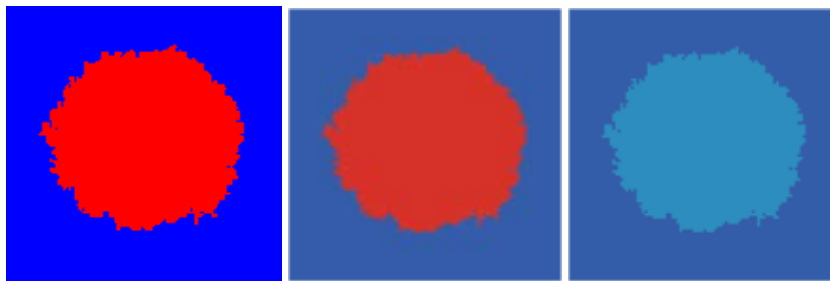


b) Notice in the model above on the left how the red color stands for cold (since it is at the bottom of the scale) while pink color stands for hot (at the top of the scale); this color mapping is

not intuitive. A superior mapping can be perceived on the right created with a color brewer color scheme ([Harrower & Brewer 2003](#)). This Heat Diffusion model ([Wilensky 1998b](#)) is at <http://ccl.northwestern.edu/netlogo/models/HeatDiffusion>. The original Heat Diffusion model and a revised version with color schemes can be found at <http://ccl.northwestern.edu/papers/ABMVisualizationGuidelines/HeatDiffusionRedesign>.



c) On the left, the ColorBrewer webpage (<http://colorbrewer.org>) displaying a divergent spectral color scheme used in the previous right Figure. On the right, the palette NetLogo extension (<http://ccl.northwestern.edu/papers/ABMVisualizationGuidelines/palette/>) for convenient selection and mapping of ColorBrewer Schemes in NetLogo.



d) The red versus blue color contrast in the left causes the Bezold spreading optical illusion, where a color seems to spread past its bounds. (Note that this optical illusion is better perceived on a CRT or LCD display, which renders white with luminosity, as opposed to print where white appears as a lack of pigmentation) The smallest effective difference ([Tufte 1996](#)) should be used to avoid the optical illusion, as demonstrated in the example on the right. The example on the left has red and blue saturated colors causing the optical illusion. The examples on the middle and right do not cause the illusion. The example in the middle has NetLogo blue and red base multihued colors based on ColorBrewer colors, and the image on the right has two hues of blue also based on ColorBrewer color schemes. The Rumor Mill model ([Wilensky 1998c](#)) is at <http://ccl.northwestern.edu/netlogo/models/RumorMill>.

Figure 5. Examples of different perception phenomena in visualizations

Gestalt Principles

6.1

Some Gestalt principles can help explain and predict a viewer's perception of a collection of graphical elements such as a visual representation of agents. Gestalt laws remain practical today due to their simplicity and generality, and have been applied to enhance contemporary visualizations ([Nesbitt & Friedrich 2002](#)).

6.2

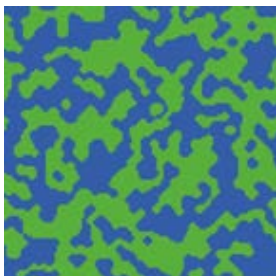
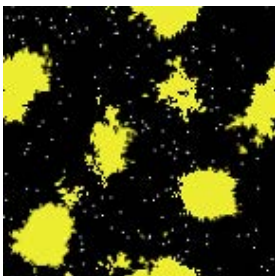

For instance, a model author can improve an ABM visualization by understanding and using the

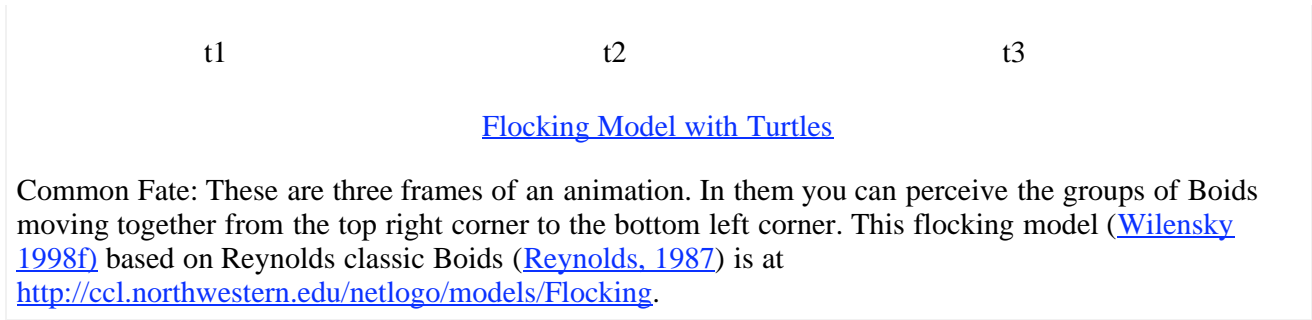
"Figure-ground segregation" mechanisms studied extensively by the Gestaltists. In NetLogo, immobile agents called 'patches' usually reside in the visualization's background, while mobile agents called 'turtles' usually reside in the foreground. A common problem in ABM visualizations arises when foreground and background are not easily distinguishable as in the model displayed in Figure 1-a at the beginning of the paper. The distinction between these two types of agents needs to be immediately and effortlessly perceived in an animation; the separation between background and foreground should be reinforced.

6.3

Gestalt principles explain how a viewer will perceive a set of disjointed visual elements by grouping them into coherent structures. These principles are relevant to ABM visualizations given that single agents can be considered as disjoint visual elements, and a collection of agents can be viewed as a coherent structure. As a result, Gestalt principles of organization provide an explanation for how the visual relations among agents can give rise to the perception of an emergent form or behavior. Below are some Gestalt principles ([Goldstein, 1999](#)) relevant to ABM visualizations:

- The principle of similarity (Figure 6-a):
"Similar things appear to be grouped together".
- The principle of proximity (Figure 6-b):
"Things that are near each other appear to be grouped together".
- The principle of common fate (Figure 6-c):
"Things that are moving together appear to be grouped together".

<p>a) Similarity</p>  <p>NetLogo Voting Model with Patches</p> <p>Color Similarity: You can distinctly perceive the green and blue groups. This Voting Model (Wilensky 1998d) is at http://ccl.northwestern.edu/netlogo/models/Voting.</p>	<p>b) Proximity</p>  <p>Termites Model with Turtles</p> <p>Distance Proximity: You can clearly perceive the yellow clusters formed by the yellow wood chips. This Termite model (Wilensky 1998e) is at http://ccl.northwestern.edu/netlogo/models/Termites.</p>
<p>c) Common Fate</p> 	






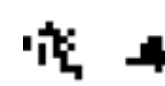

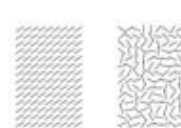




Basic Visual Variables: Position, Size, Shape, Color, Orientation & Value

7.2

In order to improve a visualization the author needs to critique his or her design. This implies having a vocabulary to describe his or her visualization and a framework to establish the alternatives for a redesign. This is why visual variables are useful. By providing a common language to deconstruct a visualization's graphic design, visual variables facilitate communication and the analysis of the graphical structure of a representation. This analysis can identify problems in a visualization and reveal plausible design alternatives.

7.2

Bertin ([1967](#); [1983](#)) defined the visual variables as position, size, shape, color, orientation, value (i.e. relative darkness or lightness of a color) and texture. These visual variables can translate directly into an agent's graphic properties. The table below (see Figure 7) contains an agent-based representation of Bertin's visual variables. In this table, rows represent the visual variables while columns represent the agent type: a single turtle, turtles, a single patch, and patches. Note that texture does not have a row as a separate visual variable, since texture requires multiple agents, it is displayed in the columns labeled 'Turtles' and 'Patches' formed by groups of agents.

Agent Type Visual Variable	Turtle	Turtles	Patch	Patches
Position				
Orientation			Does not Apply: Patches are usually represented by an upright square and polygon	
Value				









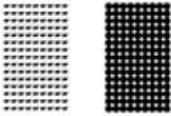
Color				
Size			Does not Apply: The size of the patch remains constant throughout the grid.	
Shape			Does not Apply: The shape of the patch remains constant throughout the grid.	Does not Apply: In NetLogo a patch is represented by a square. Few ABM toolkits support other grid shapes (e.g. hexagons)

Figure 7. Bertin's Visual Variables illustrated with NetLogo. An interactive applet, which changes each visual variable individually, can be viewed at <http://ccl.northwestern.edu/papers/ABMVisualizationGuidelines/VisualVariables/>.

7.3

While Bertin's basic ideas provide some understanding of the advantages and disadvantages of a given representational choice, it is not immediately apparent how to make use of this information in animated computational visualizations. Bertin carefully prefaced his work with the disclaimer that it was not immediately applicable to "cinematic displays", but conceived 'expressly for print' (Bertin, 1967; Bertin, 1983). As a result, the "movement" or "change" visual variable is not included in his framework. However, recent discussions of Bertin's framework (Carpendale 2003), have considered including movement as a visual variable. Additionally, in recent years, research on animated visualization has explicitly been addressed by visualization experts using cognitive science to understand how visual features should be used in real time multimodal displays (Healey et al, 1996)



Basic Visual Variables Properties: Selective, Associative, Quantitative, Order & Length

8.1

In this section we describe another part of Bertin's framework in order to support the selection of appropriate visual variables for an ABM visualization. It is not always straightforward to find the best way of representing a model's variable when it can be signified by value, color, shape, size, or orientation. Thus, understanding the characteristics of each visual variable can assist the modeler in finding the most appropriate visual variable to support a given task.

8.2

The characteristics of visual variables can indicate how a variable may aid or hinder the performance of a task. Below we present Bertin's visual variable characteristics within the context of agent based modeling:

- **Selective:** A visual variable is selective when an agent can be immediately distinguished from its surrounding agents.
- **Associative:** A visual variable is associative when groups of agents can be instantly perceived.
- **Quantitative:** A visual variable is quantitative when it allows a quick estimation of numerical ratios among agents.

- **Order:** A visual variable is ordered when it affords an unambiguous and swift comparison, or ordering of the agents. **Length:** The length of a visual variable describes how many distinct values it supports, and therefore how much information the variable can convey.

8.3

The first four characteristics (selective, associative, quantitative, order) can help the model author decide if a visual variable is suitable to accomplish a given task (target detection, cluster detection, ratio estimation, value distinction). For example, a square can be difficult to locate in a group of circles as seen in Figure 8-c1. The last characteristic (length) assists the model author in determining if a visual variable has sufficient resolution to represent a model's variable. If a visual variable does not have the necessary length to represent the model's variable none of the tasks above can be accomplished due to a resolution mismatch.

8.5

Illustrations, developed in NetLogo, of the visual variables characteristics can be viewed in Figures 8 and 9. They were inspired by Figures in Bertin (1967) and Carpendale's (2003) research. Figure 8 contains turtle agents and Figure 9 contains patch agents. Turtle agents are usually represented by circles in order to keep the examples devoid of unnecessary interferences, except in the orientation or shape rows where the heading or shape of the agent is fundamental.

8.6

In Figures 8 and 9, the columns hold five different characteristics of visual variables (selectivity, associativity, quantitative, order, and length), and the rows hold six different visual variables (position, size, shape, value, color, and orientation). The first two columns contain illustrations depicting the selectivity and associativity of each visual variable in a 2D space. The agents are arranged randomly for performing detection tasks such as boundary and target detection. These illustrations contain 50 agents, which we consider a reasonable amount for a detection task. The reader can change the number of agents and other features in an interactive applet available online (link in Figure 8). In the following three columns, the agents are arranged linearly for ratio estimation (quantitative), value comparison (order) and the depiction of their length (length).

8.7

In Figures 8 and 9, the green checkmark ✓ means that the visual variable easily supports the characteristic. For example, in the first row and column, we can observe that the position visual variable supports selectivity. The orange tilde ~ means that the visual variable may support the characteristic but care should be taken to assure it does. Finally, the cross ✗ means that the visual variable does not support the characteristic well and its use should be avoided whenever possible.

Description of a Turtle's Visual Variable Characteristics.

8.8

In this section we will discuss the characteristics of each visual variable: position, shape, size, value, color and orientation. This section should be read while referring to Figure 8 and 9 or the applet at <http://ccl.northwestern.edu/papers/ABMVisualizationGuidelines/VisualVariables/>.

8.9

In the first row, position is selective (Figure 8-a1), thus it allows pinpointing every single agent with a different location. Additionally, it is associative (Figure 8-a2) since you can perceive immediately clusters of agent. In agent-based models, the position does not usually represent a value, instead it represents a geometric location, and thus we decided that it could not be used as either a quantitative (Figure 8-a3) or ordered (Figure 8-a4) visual variable. The length (Figure 8-a5) of position, or location, is limited to the resolution and size of the screen.

8.10

In the second row, size is selective (Figure 8-b1), a reasonably larger agent can be singled out immediately, while a smaller agent can be noticed but it takes considerably more time. Two groups with different sizes can be discerned independent of their position; thus size supports associativity (Figure 8-b2). Size can be quantitative (Figure 8-b3) but it is easier to estimate the ratio between two

different lengths than between two different areas. The size of agents is easily perceived as ordered (Figure 8-b4) and has a length (Figure 8-b5) restricted to the resolution and size of the screen.

8.11

In the third row, shape can be selective (Figure 8-c1) depending on the particular shapes; a circle is easily distinguished from a line but not from a square. The same concept applies for associativity (Figure 8-c2), a group of circles is easily distinguished from a group of lines but not from a group of squares. It does not make any sense to compare the value of two different shapes, thus shape is not quantitative (Figure 8-c3) and also cannot be ordered (Figure 8-c4). However, shape has for our purposes an infinite length (Figure 8-c5) since it can take unlimited forms.

8.12

In the fourth row, value is selective (Figure 8-d1), a black agent is quickly found among gray agents. Darker agents can be easily distinguished from lighter agents thus value is associative (Figure 8-d2). However, comparing different shades is difficult; it is hard to say that an agent is twice as dark as another agent thus value is not quantitative (Figure 8-d3). On the other hand, ordering gray levels is straightforward and can be easily achieved, thus value is ordered (Figure 8-d4). Value appears to have an unlimited length (Figure 8-d5), even if a normal display has only 256 shades of gray it seems unlimited to the eye, a viewer can distinguish fewer than 256 shades ([Ware 2002](#)).

8.13

In the fifth row, color is selective (Figure 8-e1); a red agent can be quickly distinguished among black agents. In addition, agents sharing a hue can be easily separated from another hue, thus color is associative (Figure 8-e2). However, it is difficult to compare colors: you cannot quickly quantify how much orange color an agent has, thus color is not quantitative (Figure 8-e3). Colors can usually not be ordered (Figure 8-e4) on hue alone. Note that combinations involving three variables: hue, saturation, and value can be ordered, such as the Color Brewer schemes ([Harrower 2003](#)). Hue appears to have an unlimited length (Figure 8-e5), a contemporary computer display has at least 65k colors which seems unlimited to the eye.

8.14

In the sixth row, orientation is selective (Figure 8-f1); a viewer can detect immediately an agent with a different heading as long as it is a reasonably elongated shape. Also, groups with two different orientations can be associated (Figure 8-f2). However, it is best to consider orientation as not a quantitative (Figure 8-f3) or orderable (Figure 8-f4) visual variable. Even if most ABM toolkits have clear angle and heading conventions, orientations cannot be easily compared due to many difficulties, such as angles wrapping around zero. The length of orientation (Figure 8-f5) seems to be infinite and only limited by the resolution of the screen. For a detailed discussion consult ([Ware 2000](#)).

Property Visual Variable	Selective	Associative	Quantitative	Order	Length
Position	a1	a2	a3 Does not Apply Agent's position does not usually represent comparable lengths.	a4 Does not Apply Agent's position usually represents a location not a value.	a5
Size	b1	b2	b3	b4	b5

Shape	c1 	c2 	c3 Does not Apply An agent's shape does not usually represent a quantity.	c4 	c5
Value	d1 	d2 	d3 	d4 	d5
Color	e1 	e2 	e3 	e4 	e5

Figure 8. Visual Variables Characteristics of Turtles. An interactive applet can be viewed at <http://ccl.northwestern.edu/papers/ABMVisualizationGuidelines/VisualVariablesProperties/> .

Description of Patches' Visual Variable Characteristics

8.15

Patches lack many of the turtles' visual variables such as size, shape, and orientation. Indeed, in NetLogo all patches have the same size, they are all square, and since squares are not thin elongated shapes they do not have an orientation. Nonetheless, patches share some of the turtle's characteristics such as position, value and color, but a patch's position is somewhat different than a turtle's location. For example, a patch's position tends to be more selective and associative than in turtles, since their discrete positioning forbids patches from overlapping and since neighboring patches are located back to back. The length of patch visual variables is restricted to multiples of the patch size.

8.16

For the sake of brevity, we will not describe thoroughly each cell as we did on the previous subsection, the reader can apply an the same reasoning we used for the turtles to the patches.

Property	Selective	Associative	Quantitative	Order	Length
Position	a1 	a2 	a3 Does not Apply Agent's position does not usually represent comparable lengths.	a4 Does not Apply Agent's position usually represents a location not a value.	a5
Value	b1 	b2 	b3 	b4 	b5

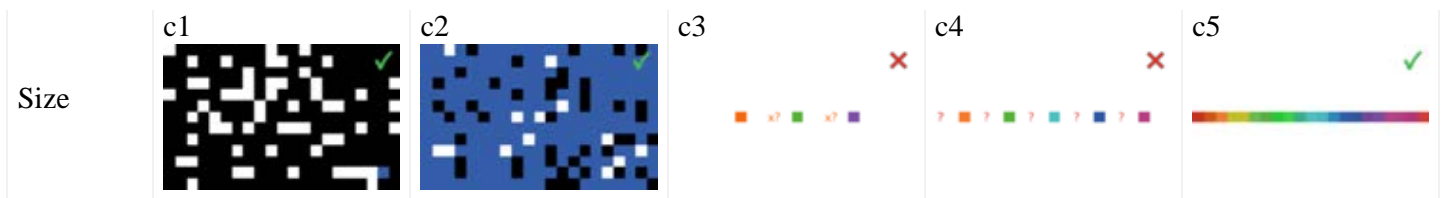


Figure 9. Visual Variable Characteristics of Patches. An interactive applet can be viewed at <http://ccl.northwestern.edu/papers/ABMVisualizationGuidelines/VisualVariablesProperties/>.

Other Visual Variables

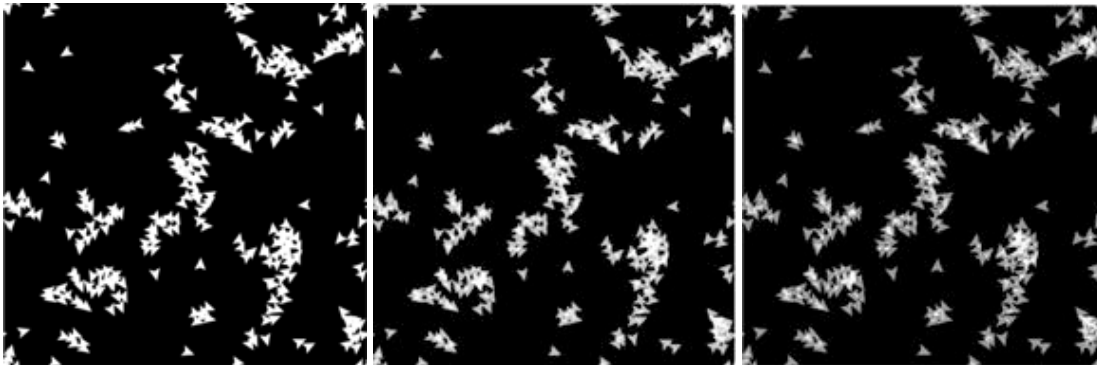
Transparency, Texture, Crispness & Resolution

9.1

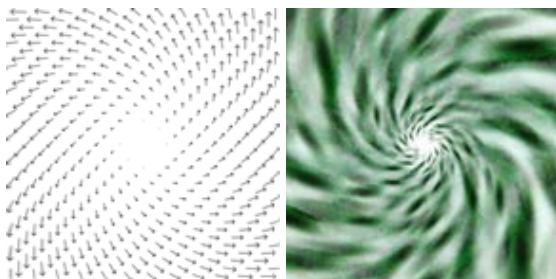
We previously discussed some of the basic visual variables mentioned by Jacques Bertin in 'The Semiology of Graphics' ([Bertin 1967](#)). In this section, we will briefly mention other more sophisticated visual variables such as transparency, texture, crispness and resolution. A thorough treatment of these visual variables is beyond the scope of this paper. We refer the reader to textbooks by MacEachren ([1995](#)) and Ware ([2002](#)) for an in depth discussion of these variables, though their treatment is not specific to ABM visualizations.

9.2

Variables such as transparency, texture, crispness and resolution can be valuable in ABM visualizations. Below we present a few examples using these visual variables in ABM visualization. Transparency can assist in the perception of agents covering or overlapping other agents as seen in Figure 10-a. Texture can increase the number of variables that can be simultaneously displayed (though it considerably decreases the resolution as seen in Figure 10-b ([Julesz 1986](#) ; [Ware 2002](#))). Crispness allows the user to focus on a particular agent while maintaining its context -- a technique known as semantic depth of field Figure as seen in 10-c ([Kosara 2001](#)). Also, resolution can be decreased by blurring each individual agent to allow the user to easily perceive the macro behavior of the model as in Figure 10-d.



a) Example of agents with transparency in the middle and right images. (Colby 1991).



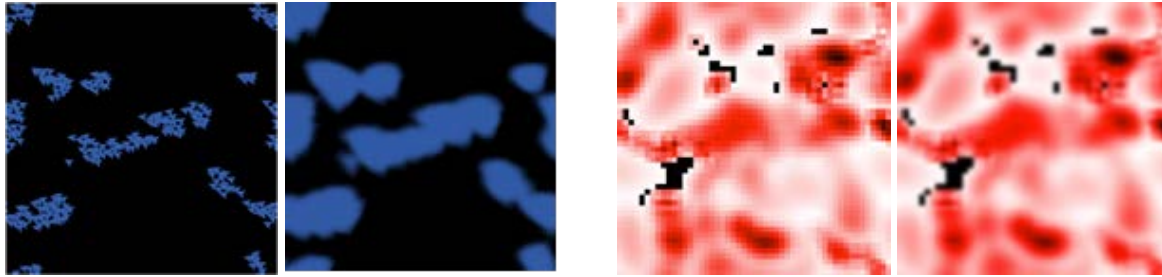
a) Example of a vector field generated in NetLogo



b) Example highlighting agent micro behavior.

with arrows on the left and with texture on the right generated by agent with textons (Jules 1986; Ware 1995).

Focus on a single agent with a halo on the left; while on the right we blurred the individual agents.



d) Examples of agents with decreased resolution to enable easy observation of emergent macro behavior

Figure 10. Examples of transparency, textures, resolution, and crispness in ABM visualizations. (We created these images using experimental NetLogo builds, except for Figure 10-d on the right where we blurred a screenshot with a graphics program)

Change

9.3

ABM visualizations are intrinsically animated since change over time is the most straightforward way to represent behavior. In this section, we will discuss issues that frequently arise while updating ABM visualizations. A thorough treatment of dynamic visual variables is beyond the scope of this paper. For a detailed definition of these visual variables we refer the readers to ([MacEachren, 1995](#)) who defines six dynamic variables: duration, rate of change, order, display date, frequency and synchronization. Note that their treatment is not specific to ABM visualizations, but rather targeted to maps.

9.4

Smooth graphical transitions are quite challenging to update for ABM toolkits such as NetLogo where an interactive Agent Based Model is executed in real time. The challenge is to balance the use of CPU time between displaying graphics and performing model calculations. If CPU time is devoted exclusively to graphics updates, the model's calculations grind to a halt. On the other hand, if the computational effort is solely dedicated to the model's calculations the graphics slow down to an unacceptable rate. Note however, that this is not an issue in information visualization where the data is readily available for display since it is already stored and does not have to be computed. This is an important distinction between information visualization and interactive ABM visualization. In information visualization most of the CPU cycles can be used to update graphics and interact with the user.

9.5

This results in two main issues that arise in ABM animation:

- Temporal aliasing, when the agents become visually hard to track since they abruptly 'jump' from one location to another instead of smoothly 'sliding' between positions. ([Korein 1983](#))
- Irregular frame rate, where the frame rate fluctuates over time depending on the complexity of the model calculations in such a way that it confuses and frustrates the user.

9.6

Spatio-temporal aliasing or non-smooth jerky motion can pose insurmountable perceptual challenges in ABM visualizations where the user is trying to keep track of the agents in the simulation. The problem of temporal aliasing has long been recognized and many anti-aliasing techniques have been developed including supersampling and motion blur. Currently, supersampling is usually the only option that the model author can implement, where he increases temporal resolution by interpolating

intermediate agent positions between two locations. A good rule of thumb is to move the object at most 1/3 of its size between each frame ([Ware 2002](#)). This criterion does not apply if the object is very small (such as 1 pixel wide) or very large. Motion blur probably requires accessing lower level graphics, which are usually inaccessible in a high level toolkit.

9.7

Irregular frame rate is due to the changing computational load at each iteration. This unpredictable rate can confuse and frustrate the viewer. If the computational load fluctuates due to a changing number of agents, the model author can try to stabilize updates by intentionally slowing down the model to the lowest possible frame rate.

9.8

Ideally, agents should be bounded with physical restrictions (viscosity, inertia, etc ..) in such a way that agents change their position or appearance in a smooth predictable manner. In the future, these physical restrictions and other artifacts such as the previously mentioned (temporal aliasing and regular frame rate) should be automatically handled by toolkits. New developments in hardware and software will implement this functionality. For example in hardware, multi processors systems or GPUs (Graphical Processor Unit) can be solely dedicated to graphics and perform the necessary computations. Additionally in software, new animation frameworks are managing individual elements on a frame-by-frame basis. In these new frameworks, a smooth animation can be created by specifying the start and end position -- the libraries calculate the intermediate locations for a smooth transition.

9.9

There are however many visualization aspects that cannot be automated, for example facilitating a particular user such as tracking, estimation or segregation. Additionally, the model author should prevent visual interferences such as those discussed in the next section.



Cognitive Science Visualization Principles

10.1

Cognitive Science has elucidated many principles of effective, as well as misleading, visualizations. Here we focus on two cognitive phenomena the user should be concerned with when creating an ABM visualization: pre-attentive visual processing and visual interference. Creating visualizations that can be pre-attentively processed and are free of visual interference enables a viewer to perform fast, automatic and spatially parallel operations to understand the visualization. Pre-attentive operations can detect visual features such as intensity, hue, and orientation and assist in tasks such as boundary detection and high-speed target tracking ([Healey 1995; 1996](#)). However, pre-attentive operations can be impeded by visual interference. For example, visual interference can occur when a chosen visual variable conflicts with another visual variable. Another example of visual interference occurs when a selected visual variable is so dominant that it makes the other visual variables imperceptible.

Pre-attentive Visual Processing

10.2

Pre-attentive visual processing enables the fast and scalable perception of the features of a visualization. It allows the viewer to process animations containing a large number of graphic elements, as is often the case in ABM animations. Pre-attentive visual tasks can usually be accomplished in less than a quarter of a second ([Ware 2004](#)), independent of the number of graphic elements involved, which allows the viewer to process high-speed, information-rich ABM animations.

10.3

In order to construct an ABM visualization that can be processed pre-attentively, the modeler must pinpoint the visual tasks he requires from the viewers. By doing so, the modeler can select the optimal visual features to accomplish the task and construct a visualization that allows the viewer to

accomplish the task pre-attentively. For example, the visualization's author can select color or size to perform pre-attentive visual tasks, shown in Figure 11, such as target detection, boundary detection, region tracking, counting and estimation (Healey 2006).

10.4

Pinpointing the task depends on the phenomena the modeler wishes the viewer to focus on. These phenomena can be either macro- or micro-behaviors of the model. Perceiving macro-behavior involves tasks such as boundary detection or region tracking. Both of these tasks assist the viewer in discerning and tracking clusters of agents. Macro-behavior perception can be supported by other visual tasks as well, such as estimation, which facilitates the approximation of the quantity of agents sharing a visual feature (independent of their position and quantity).

10.5

Perceiving micro-behavior involves tasks such as target detection, where the user must focus on the behavior of a single agent amongst all the agents. Tracking a single agent can be quite a challenge in a flock of agents where the user is distracted by the movement and direction changes of other agents.

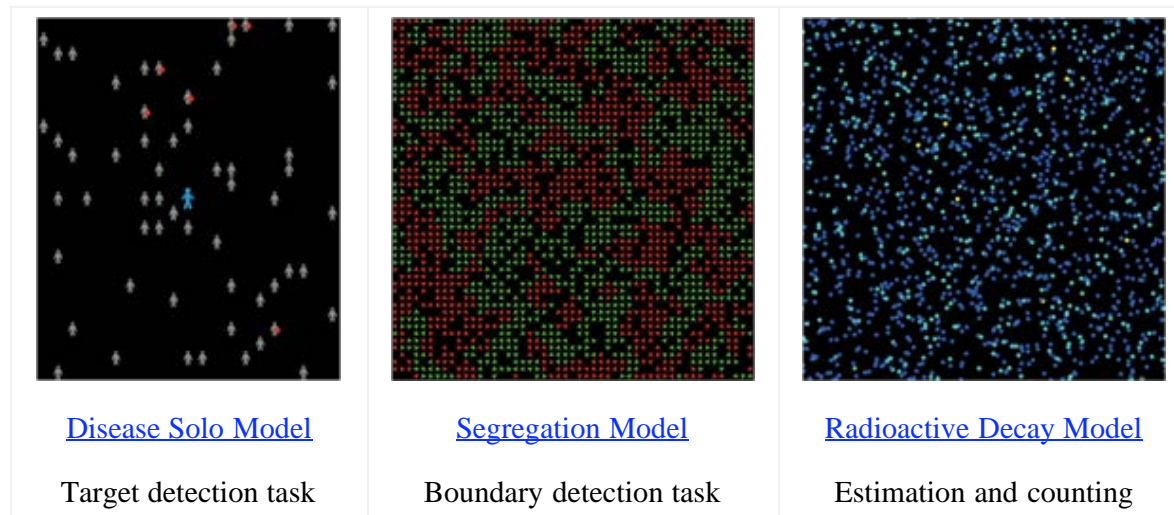


Figure 11. Examples of target detection, boundary detection, and counting & estimation tasks in NetLogo Models. The Disease Solo Model (Wilensky 2005) is at <http://ccl.northwestern.edu/netlogo/models/DiseaseSolo>. The Segregation Model (Wilensky 1998g) is at <http://ccl.northwestern.edu/netlogo/models/Segregation>. The Radioactive Decay Model (Wilensky 1998h) is at <http://ccl.northwestern.edu/netlogo/models/Decay>

10.6

Pre-attentive perception demands different design priorities in dynamic animations than in static images. For example, in static images, feature comparison takes place in space, thus spatial resolution plays an important role. However in dynamic animations feature comparison takes place in time. Thus, the temporal resolution is more important than the spatial resolution, since the changes in time are crucial to studying behavior. So high spatial resolution, which is helpful in static images in order to provide a large information density, should usually be avoided in dynamic visualizations in order to facilitate a rapid cognition for each frame of the animation.

10.7

Once the modeler has pinpointed the task necessary to highlight a specific phenomenon, he can begin by ensuring there is no visual interference confusing the user. If visual interference exists, he should change or remove the visual variables causing the visual interference. In the next section we will discuss some examples of visual interference in ABM visualizations.

Visual interference

10.8

Visual interference, as described by Callaghan (1989), takes place when a pre-attentive task is

hindered by the concurrent use of interfering visual features. For example, estimation is hampered in hue-luminance interference, since luminosity is prioritized over hue by the low-level visual processing system (see Figure 12-a and 12b).

10.9

This particular interference is caused by a "feature hierarchy that appears to exist in the visual system" (Healey 2006). Other interferences emanating from the feature hierarchy are: color-over-shape, hue-on-form and hue-on-texture. Examples of each of these types of interference can be observed in Christopher's Healey's comprehensive review (<http://www.csc.ncsu.edu/faculty/healey/PP/index.html>). Additionally, these three interferences can be observed in Figure 12-c and 12-d,

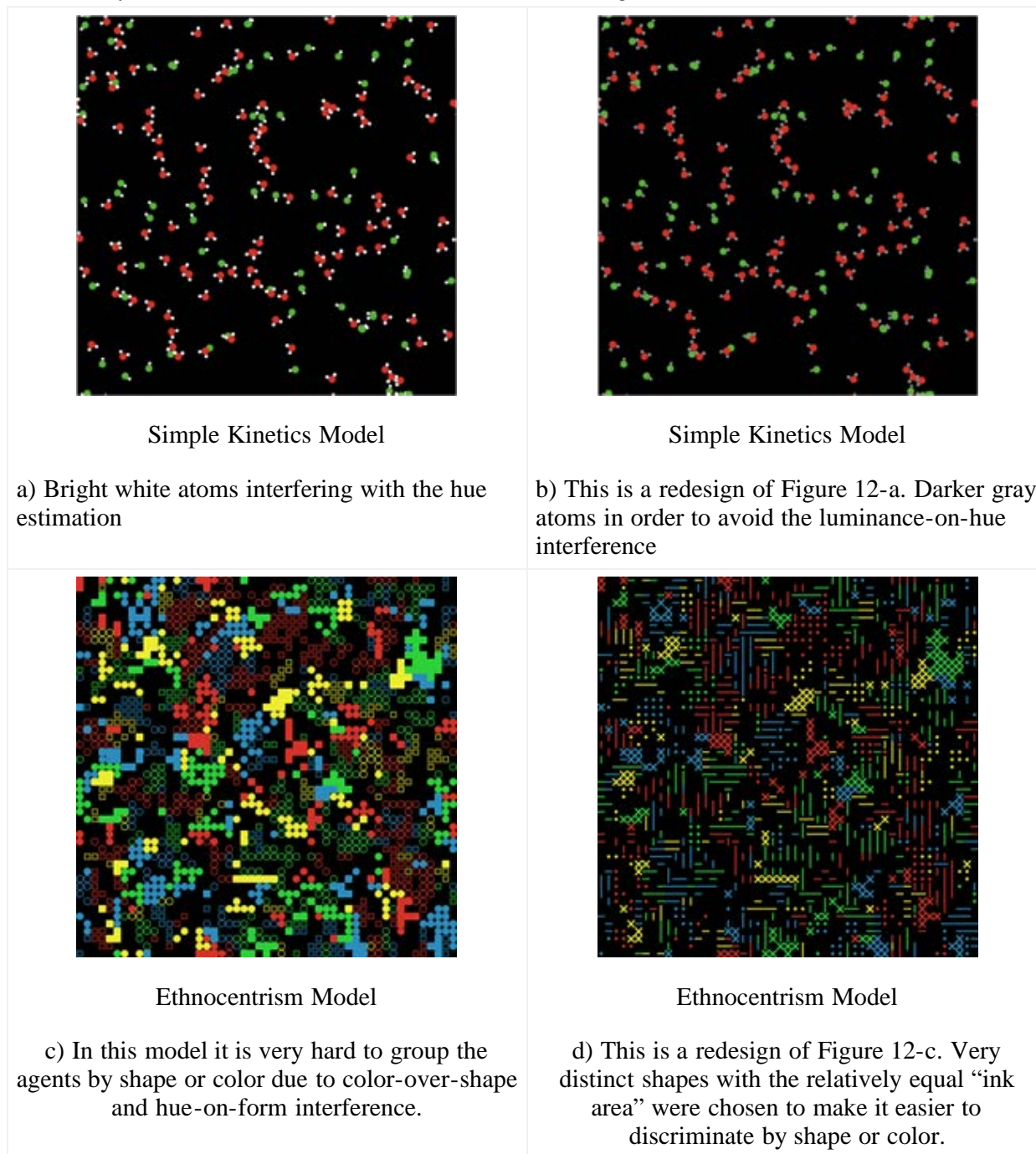


Figure 12. In the ABM visualizations a) and b), the key visual feature to distinguish is hue, which allows the viewer to estimate the number of red and green molecules. However, on the left, the bright white atoms interfere with the hue estimation. Changing the bright white atoms to darker gray atoms

remedies the luminance-on-hue interference. Note that this interference is mostly perceived on a CRT or LCD display, which renders white with luminosity, as opposed to print where white appears as a lack of pigmentation. The Simple Kinetics 1 ([Wilensky 1998i](#)) Model is at <http://ccl.northwestern.edu/netlogo/models/SimpleKinetics1>. In the Figures 12-c and 12-d there are several key features to distinguish, mainly shape and color. This model contains four shapes: squares, hollow squares, circles, and hollow circles. They were replaced by crosses, dots, horizontal lines, and vertical lines. It was originally unfeasible to group the squares (filled square and hollow squares) due to the difference of luminosity between the hollow and filled shapes. The creation of textures allows the viewer to distinguish color and shape independently. However, this redesign still suffers from other interferences such as hue on texture. This redesign does not solve all the issues of this model, however it is an improvement. The NetLogo Ethnocentrism model ([Wilensky 2006](#)) is at <http://ccl.northwestern.edu/papers/ABMVisualizationGuidelines/EthnocentrismUserStudy/>.

10.10

Visual interference can occur in agent-based models when the agent's variables are mapped to visual features without considering whether they interfere or not. By being aware of well-known visual interference patterns, the modeler can determine when interference is occurring and prevent it. The perception of the main variables in agent-based visualization should not be hampered when displaying secondary variables. Whenever possible, the modeler should display the model's variables of interest in accordance with their perceptual salience. The perceptual salience of a visual feature depends on the task, but for many tasks the most salient visual features tend to be luminance and hue, followed by size, length orientation and shape ([Healey 2006](#)).



Design Objectives

11.1

Design principles are not sufficient to improve a visualization, certain objectives need to be established in order to guide, when and how, design principles enhance the visualization. In this section we present the intent for applying the previously stated principles: to simplify, to emphasize and to explain.

11.2

Design criteria have been studied at length by many visualization researchers such as Tufte. However, the criterion for improving a visualization changes depending upon the type of visualization and the task at hand. Unlike mainstream visualizations, ABM representations display a behavior mostly driven by rules rather than data. Thus, dynamic ABM visualizations need to fulfill different requirements than usual static descriptive depictions.

11.3

In static visualizations, where the viewer has the option of carefully studying the motionless image, high data density is recommended to provide as much information as possible. However, within an ABM animation, a low data density is recommended to allow the viewer to follow changes in the visualization. ABM animations ought to be simple and emphasize their key variables to facilitate pre-attentive processing in spite of containing a multitude of changing graphical elements.

11.4

Another difference is that ABM visualizations are often unique and therefore they cannot be understood as easily as common, conventional visualizations such as histograms or pie charts. Most ABM visualizations are custom designed to depict a particular phenomenon, and consequently, they are not intuitively understood. Many ABM visualizations share common graphical elements, but these elements can represent different objects or properties. For example, a circular agent may represent a particle in one visualization or a node in a network in another visualization. These visualizations that lack convention can only be understood if the author designs visual explanations that make explicit the point of the model.

Design Approach

11.5

Below, we describe several steps for incorporating some of the design principles mentioned previously: simplify, emphasize and explain. These steps are intrinsically generic due to the open-ended nature of ABM visualizations. However, we believe this approach can offer some structure for iteratively redesigning an ABM visualization. Note that, depending on the type of visualization the modeler can find more rigorous frameworks that can also apply to their design. For example, (Jones 1998) provides a detailed map-making approach in his textbook 'Geographical Information Systems and Computer Cartography'. In the Figure 13 below we summarize an approach to redesigning an ABM visualization

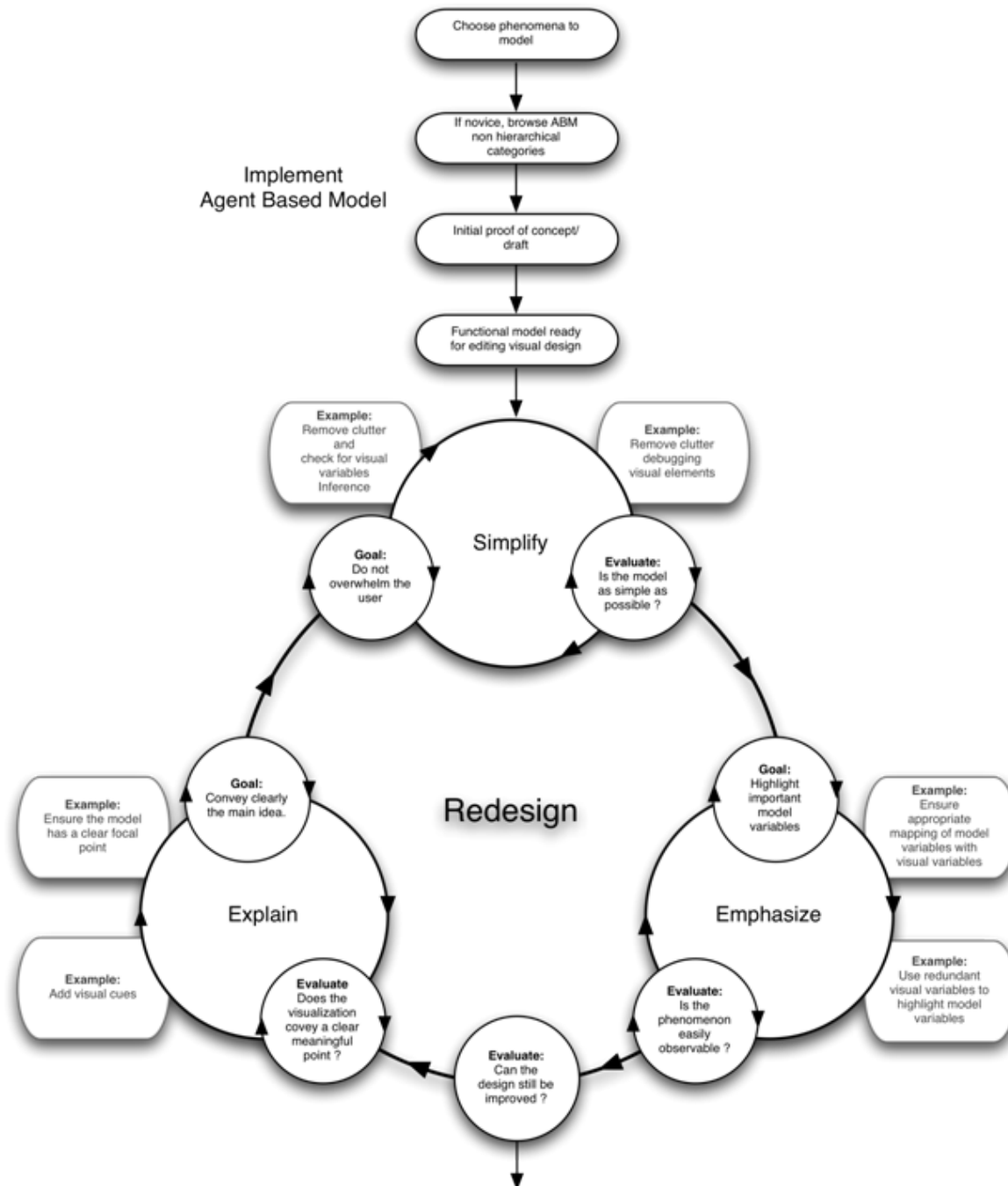


Figure 13. Generic design workflow of an ABM visualization. Initial implementation of the model followed by an iterative redesign composed by 3 steps Simplify, Emphasize, and Explain.

Simplify

11.6

Simplifying ABM visualization enables the user to concentrate on the model's main behaviors without unnecessary distractions. One of the first enhancements to an ABM visualization is to remove clutter or in Tufte's words "Chart Junk" (Tufte, 1983). Clutter in ABM is usually caused by dispensable graphic elements created for testing and debugging purposes during the model's conception. A visualization displaying many variables needed to debug a model may not be optimal for viewing the model's important behaviors. The viewer can be overwhelmed and confused by the visual variables useful to the model's author, but devoid of meaning for the intended audience.

11.7

This tends to happen especially in development environments where graphical elements can be added easily. Thus, the modeler would benefit by uncluttering his / her visualization before he presents it to other users. Removing construction and debugging graphical elements enables the observer to easily perceive a certain behavior or pattern.

11.8

When there are a large number of simultaneous behaviors and graphical elements to observe, the viewer should have the option to hide graphical elements until they reach a set of cognitively manageable features. For instance, in GIS environments, this is accomplished through the use of layers. Users can turn layers off and on in order to allow a GIS user to show and hide information in a map as they examine the data.

11.9

Another similar way to simplify a visualization is to divide it into two or more visualizations, where each one renders different model variables, such as the Rumor Mill model below in Figure 14.

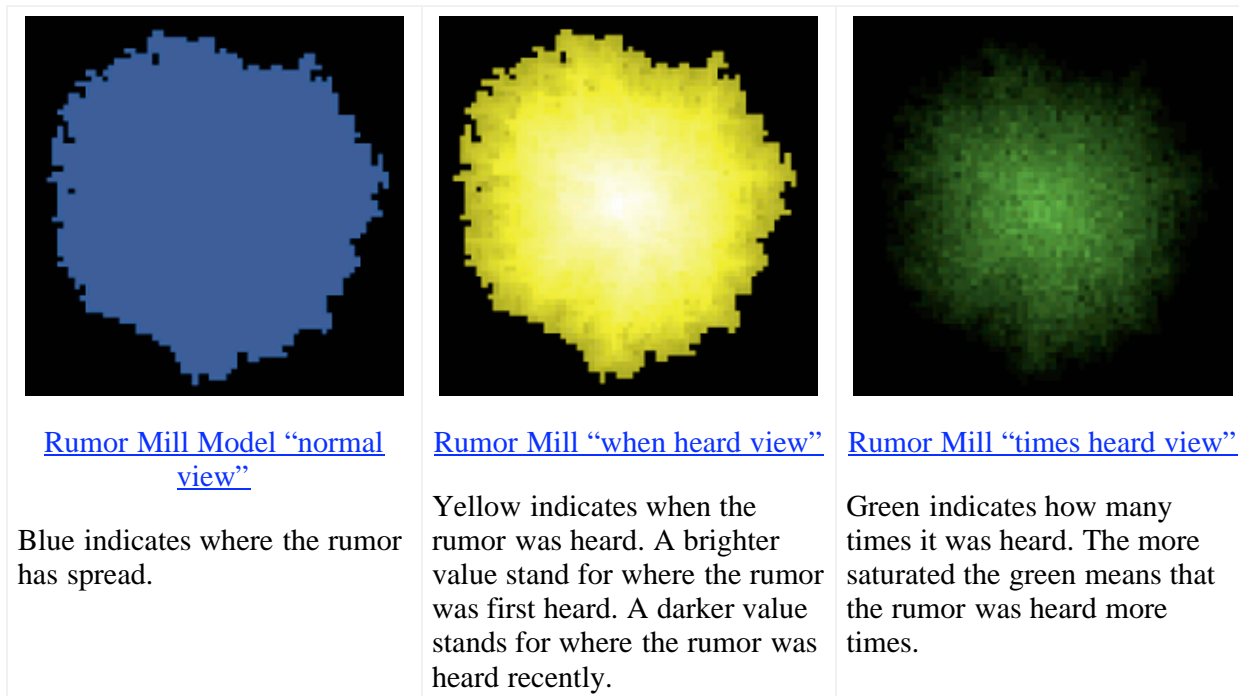


Figure 14. Several views of the Rumor Mill Model: This model shows how a rumor spreads. The rumor spreads when a person (represented by a patch) tells a rumor to a neighbor (represented by an adjacent patch). At each time step, every person who knows the rumor randomly chooses a neighbor to tell the rumor to diffuse. You can view the Rumor Mill Model (Wilensky 1998c) at <http://ccl.northwestern.edu/netlogo/models/RumorMill>.

Emphasize

11.10

In animations the user must instantly perceive how the system behaves frame by frame. This perception can only be achieved using pre-attentive processing. For instance, the user does not have time to think about selecting a target, he ought to be able to select and follow a target using his low level cognitive abilities. One technique used to support pre-attentive processing is to represent a model variable using several redundant visual variables.

11.11

For example, as shown in Figure 15, the association of the orientation of agents in the left-hand image can be enhanced by mapping the orientation of the agent to a hue as shown in the center image. The perception of orientation can further be emphasized by use of elongated shapes, as illustrated in the right image.

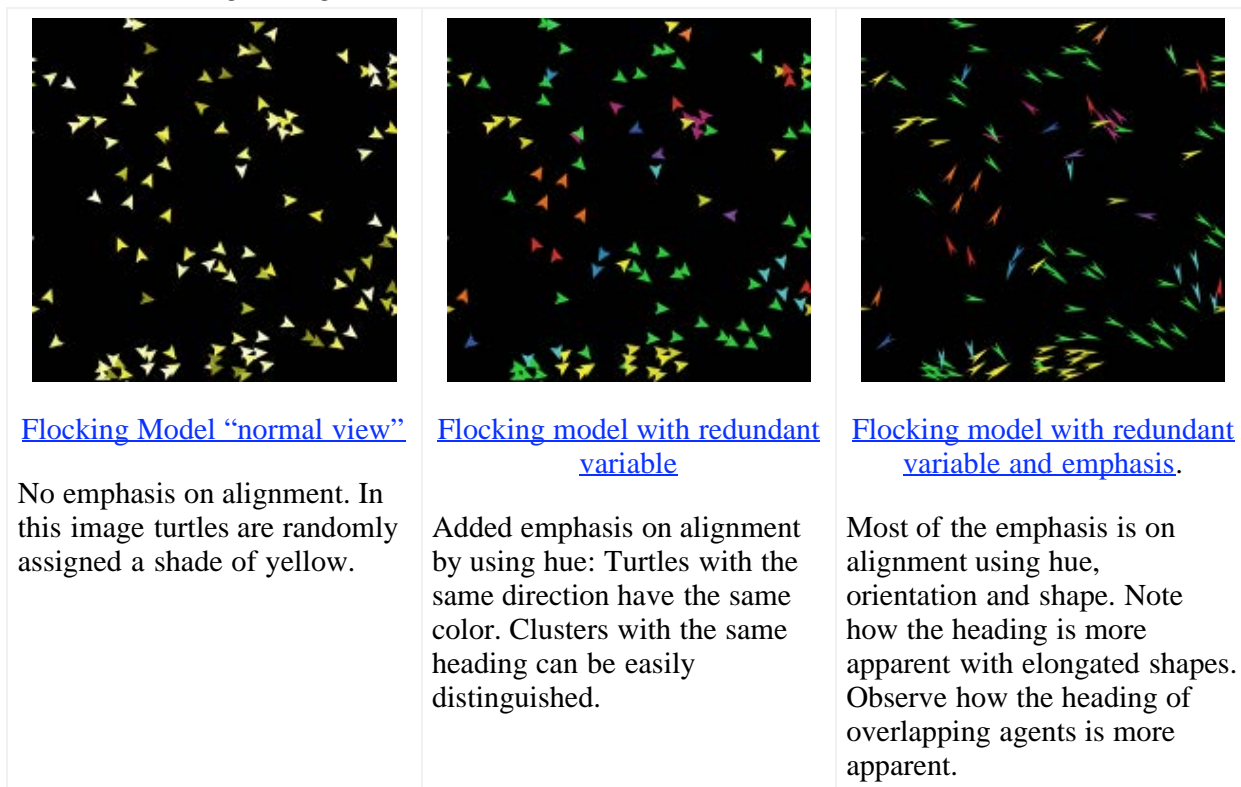


Figure 15. Redundant visual features of the flock alignment. This redesigned flocking model is at <http://ccl.northwestern.edu/papers/ABMVisualizationGuidelines/FlockingRedesign/>

11.12

The use of several visual variables to represent a single model variable is a common technique. Redundant visual variables can increase the legibility and expressiveness of interactive maps for visual data exploration ([Andrienko, 1999](#)). Thus, using redundant visual variables in agent based models, should facilitate the cognition of information within ABM animation.

Explain

11.13

ABM visualizations usually have only one viewing mode depicting the macro-behavior of the model. However, the audience should have the option of viewing the agents' micro-behaviors as well. Observing the micro-behavior can provide a more meaningful visualization by making the rules driving the model readily available to the audience. Usually, the micro-behavior is clearly understood by the modeler since he went through the process of developing the model. He observed the rules for a single agent when he authored the model. However, this is not the case for a viewer who is observing the model for the first time. Without knowledge of the micro-behavior that produces the

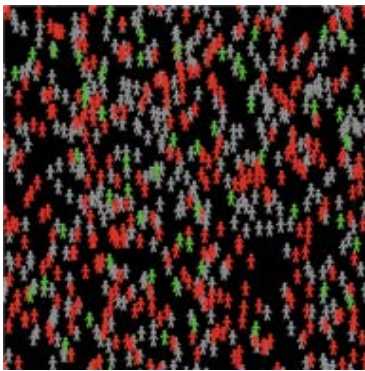
macro-behavior, first time users can be overwhelmed by the movement of many agents.

11.14

Thus, the user should have the option of changing the visualization in order to observe specific behaviors such as the behavior of a variable at either a system-wide or individual agent level. A simple way to allow the viewer to observe the agents micro behavior, is to hide all agents except the ones that will be used to illustrate the micro-behavior. In this manner, we can suppress unnecessary moving graphical elements in the visualization that distract the viewer. This operation can be easy or quite difficult to implement depending upon the model's sophistication, the viewer's familiarity with the model, and the language primitives of the modeling toolkit.

11.15

Furthermore, the visualization of a single agent allows for the depiction/representation of patterns that would be impossible to follow with all the agents on screen. For example in Figure 16 on the right, the viewer can see the displacement of a single agent in time by asking the agent to leave a visible trail. However, it would be impossible to use this same technique with multiple agents, since the trails would merge and become indistinguishable. This is distinct from modeling a single agent, all the agents are present in the model but only one agent, the spatially proximate agents and their interactions are visible.



[Virus Model "normal view"](#)

Virus model in a macro behavior view.



[Virus Model "micro-behavior View"](#)

Virus model in a micro behavior view. You can observe the green trail of a healthy individual, which becomes sick when the trail changes to red. Additionally, you can see the individual who transmitted the virus in a darker shade of red linked by a line.

Figure 16. It is difficult to observe the micro-behavior in the left image, while in the right image, micro-behavior is explicitly shown: an agent is sick, becomes healthy, and has an offspring.
<http://ccl.northwestern.edu/papers/ABMVisualizationGuidelines/VirusUserStudy> .

Redesign Example

12.1

The intent of this section is to convey, through a simple example, the thought process behind the redesign of an ABM visualization. This example will take advantage of the various concepts we have discussed earlier in this paper and present them in a systematic, applied manner.

12.2

The model below is implemented in NetLogo and replicates the conservation of electric current passing through two different materials. It was created as part of the NetLogo Investigations in Electromagnetism project (NIELS) ([Sengupta 2006](#)). NIELS develops ABM curricular units in physics (electrostatics, electricity and magnetism) intended for high school, middle school and

undergraduate students. We will describe modifications performed during the redesign of this electric conduction model, which was conceived to provide a conceptual understanding of the resistance phenomenon.

12.3

This redesign was carried out as part of our research for this paper. In a five-hour session the authors of this paper provided feedback and suggestions to the model's first author. The modeler incorporated some of the proposed improvements and devised new enhancements of his own based on the feedback. The redesign continued with minor consultations over a few weeks after the initial session.

<http://ccl.northwestern.edu/papers/ABMVisualizationGuidelines/SeriesCircuitUserStudy>

12.4

In the initial visualization below, electrons are represented by red rings filled with a black disk as seen in Figure 17. The electrons move horizontally from the right negative terminal to the left positive terminal (blue and red backgrounds) over two different conducting materials (yellow and black backgrounds). Their movement does not perfectly follow a straight line; it incorporates a small vibration representing a minor random component.

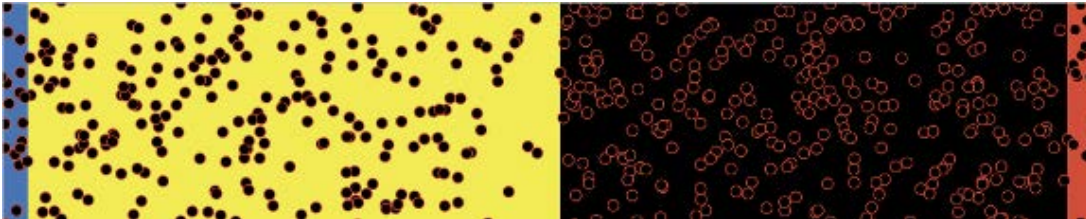


Figure 17. Initial visualization redesign of Series Circuit Model

12.5

An unintended consequence of having electrons as black disks with red rings is that they are distinguished using two different visual cognitive mechanisms. The first mechanism to identify an electron is a change in value (intensity) of the dots. In this case electrons are perceived as black dots. The second mechanism to recognize the electrons is closure ([Ware 2004](#)); in this case the closed shape of the rings allows the model user to discern electrons. The electrons appear as red rings over the black background. This dual perceptual mechanism for the same entity causes an unnecessary cognitive load for the viewer. This can be easily resolved by filling the electrons with a solid color, hence assigning them a fixed representation regardless of the changing background color. This can be seen below in Figure 8.

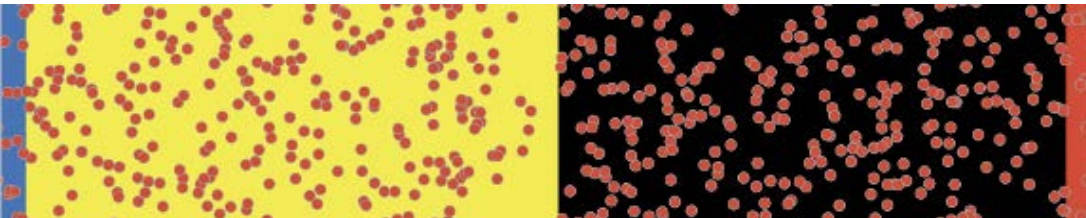


Figure 18. Foreground enhancement of Series Circuit Model

12.6

Another issue is that the conductors represented by the yellow and black backgrounds display an excessive contrast. Yellow is the hue with the highest intensity while black has the lowest intensity. However, the background should have a low contrast and low intensity in order to facilitate effortless background/foreground segregation.

12.7

Also, the electrons are too large, causing them to be perceived as regions that intersect rather than individual points. Representing the electrons as small bright points as seen in Figure 19 can facilitate the counting and estimation task, since you do not have to process their intersections when they overlap.

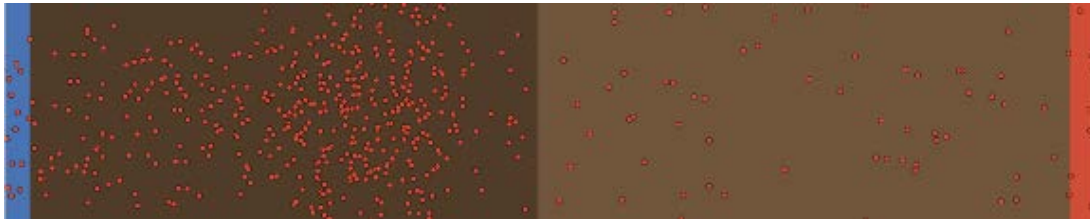


Figure 19. Foreground enhancement of Series Circuit Model

12.8

Changing the background conductor's color to similar shades of brown lowers the background's contrast and intensity. Additionally, the reduced electron size allows the viewer to perceive them as a distribution of dots rather than overlapping circles. Independently, notice that electrons are again perceived as black rings on the far right red background. Thus, the dual representation we wanted to avoid initially, of the electrons as rings, is still taking place.

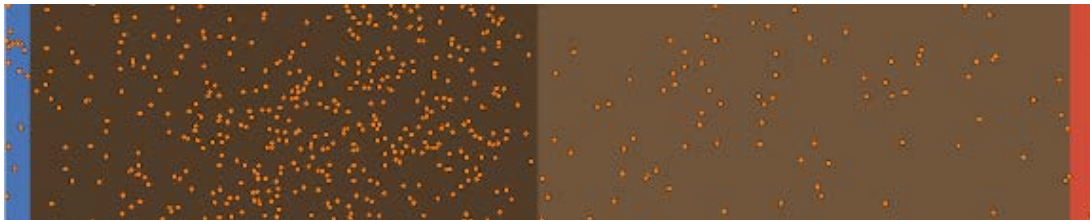


Figure 20. Foreground enhancement of Series Circuit Model

12.9

We changed the electron's color to orange to avoid this dual representation in Figure 20. However, this change did not prove useful; the orange hue is still too close to red. Notice how visible electrons are over the blue background (blue and orange are complementary) while the single electron at the bottom of the red background is virtually invisible. Moreover, the wire's brown hue is distracting; while the brown hue has the necessary contrast and intensity properties, its color does not convey any meaning and is unnecessary.

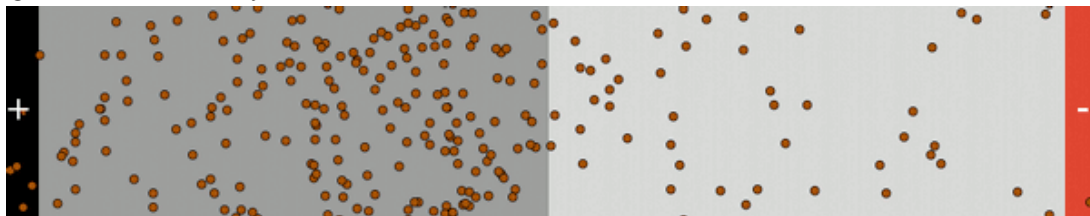


Figure 21. Final design of Series Circuit Model

12.10

In this final visualization we simplified the design in Figure 21 by using two shades of gray for the conductors. The electrons have a dark red to distinguish them from the background. And we placed labels over the terminals to make their charge explicit.

12.11

Other changes were made to improve movement, however they can only be perceived in an [animation](#). For example the motion of the electrons was jerky due to temporal aliasing. The discontinuous motion of electrons through the wire was difficult to track visually since the electrons moved with relatively large steps. In order to make the electrons easier to follow, the step that electrons took between frames was decreased to render the movement more smooth and continuous. (i.e. temporal anti-aliasing).

Conclusion

13.1

communicative and cognitively efficient ABM visualizations. We began by presenting a brief classification of ABM visualizations categorizing them into conventional, structured and unstructured visualizations, which can be used as a basis for finding visualizations that are useful as a starting point for a new visualization. We continued by describing how established design frameworks such as visual variables and Gestalt laws can be contextualized in ABM. Then, we discussed ABM visualization design goals: to simplify, emphasize and explain. Finally, we presented an example to illustrate the process driving a redesign.

13.2

These guidelines aim to reduce the cognitive load for the viewer by facilitating the perception and cognitive processing of key behaviors in an ABM. Tools and environments that facilitate the use of color, shape and movement to represent a behavior in computational media have only become available in the last two decades. Although some of the guidelines presented in this paper are well known in the scientific visualization literature, they have not been applied to agent-based models or reached an ABM audience. In fact, design guidelines geared towards displaying information in conventional static representations such as plots, histograms, and other commonplace depictions appear in a variety of educational contexts, but in general, dynamic visualization design with animated graphics has not yet found its way into standard educational curricula.

13.3

ABM visualizations have different requirements than traditional visualizations. While traditional visualization can convey data with static graphics ABM visualization needs to convey behavior instead of data using dynamic graphics. This behavior of an agent-based model must be rendered in a perceivable and recognizable manner. In the real world, images obey the laws of nature so that variations are usually discernible to the eye and understood by the mind due to their visually continuous and predictable behavior. However, an ABM model is a computational abstraction, which can exhibit abrupt visual changes that can be hard or impossible to track or understand. To effectively communicate, the author of an ABM model visualization should be capable of identifying inefficient graphical representations and know how to redesign them. Thus, a model's author should be aware of the bounds of visual cognition and perception in order to locate these unperceivable or overwhelming effects. Then, he or she should strive to make these changes perceivable and understandable using visual design. This paper has laid out some design guidelines in order to equip authors to create better ABM visualizations.

13.4

In future research we will examine more advanced and novel tools for ABM visualization. We plan to expand the expressive range of visual variables, while taking into account the user's perceptual and cognitive capabilities. This will include features mentioned previously such as luminous transparency, textons, and spatial temporal anti-aliasing. We have already developed new dialogs and primitives to aid visual variable selection, such as the NetLogo palette extension enabling users to choose color schemes that facilitate a meaningful mapping from value to color. We plan to develop shape and texture selection assistants. We are currently testing these tools to find out if they help users improve their visualization design.

References

- ANDRIENKO, G L (1999) Interactive maps for visual dataexploration, *International Journal of Geographical Information Science*, vol. 13, issue 4, p 355.
- ALBERS, J (1975) *Interaction of Color*. Yale University Press, New Haven, CT.
- BERTIN, J (1967) *Sémiologie graphique*, Paris-Den Haag: Mouton.
- BERTIN, J (1983) *Semiology of Graphics*, Translated by W. Berg, *University of Wisconsin Press*.

- BRATH, R (1997) Metrics for effective information visualization, Proceedings of the 1997 IEEE Symposium on Information Visualization (InfoVis'97), p.108, October 18-25.
- BRATH, R M Peters and R Senior, (2005) Visualization for Communication: The Importance of Aesthetic Sizzle, *IEEE IV05*.
- BACH, M, Poloschek CM (2006) Optical Illusions. *Adv Clin Neurosci Rehab (ACNR)* 6:20–21.
- BACH, M (1997-2) 67 Optical Illusions & Visual Phenomena. Last updated 26-Nov-2006. Retrieved December 1, 2006, from Bach Micheal's web site: <http://www.michaelbach.de/ot/index.html>
- CALLAGHAN, T. C. (1989) Interference and domination in texture segregation: Hue, geometric form, and line orientation. *Percep. Psychophys.* 46, 4, 299-311.
- CARPENDALE, M S T (2003) Considering visual variables as a basis for information visualization. *Technical Report 2001-693-16*, Department of Computer Science, University of Calgary, Calgary, Canada.
- COBB T, (1997) Cognitive efficiency: Toward a revised theory of media. *Educational Technology Research & Development*, 45 (4), 21-35.
- COLBY G, Scholl L, (1991) Transparency and blur as selective cues for complex visual information, *SPIE Vol. 1460, Image Handling and Reproduction Systems Integration*, pages 114–125.
- GOLDSTEIN, E B (1999) Sensation and Perception (5th Edition). *Brooks/Cole Publishing*.
- HARROWER, M A & Brewer C A, (2003) ColorBrewer.org: An Online Tool for Selecting Color Schemes for Maps, *The Cartographic Journal* 40(1): 27-37.
- HEALEY, C G, Booth K S, & Enns, J T (1995). Visualizing Real-Time Multivariate Data Using Pre-attentive Processing *ACM Transactions on Modeling and Computer Simulation* 5, 3, 190-221.
- HEALEY, C G, Booth K S, & Enns J T (1996) High-Speed Visual Estimation Using Pre-attentive Processing *ACM Transactions on Human Computer Interaction* 3, 2, 107-135.
- HEALEY, C G, St. Amant R, & Elhaddad M. (1999) "ViA: A Perceptual Visualization Assistant." In Proceedings 28th *Applied Imagery Pattern Recognition Workshop*, (Washington, D.C., pp. 1-11.
- HEALEY, C G (2006) Perception in Visualization, Last updated 17-Jan-2007. Retrieved December 1, 2006, from Christopher Haley's web site: <http://www.csc.ncsu.edu/faculty/healey/PP/index.html> (comprehensive review updated regularly).
- ITTEN, J (1970) The Elements of Color. *Van Nostrand Reinhold Company*, New York, NY.
- JONES, C (1998) Geographical Information Systems and Computer Cartography, Longman, Harlow, England.
- JULESZ, B (1986) Texton gradients: The texton theory revisited, *Biological Cybernetics*, vol. 54, pp. 245-251
- KLEIN, J (2002) Breve: a 3D simulation environment for the simulation of decentralized systems and artificial life. Proceedings of Artificial Life VIII, the 8th International Conference on the Simulation and Synthesis of Living Systems. *The MIT Press*.
- KOSARA R, Miksch S, Hauser and H, (2001) Semantic Depth of Field, *Proc. IEEE Symp. Information*

Visualization (InfoVis2001), IEEE CS Press, LosAlamitos, Calif., pp. 97- 104.

KOREIN J, Badler N, (1983) Temporal anti-aliasing in computergenerated animation, Proceedings of the 10th annual conference on Computergraphics and interactive techniques, p.377-388, July 25-29, Detroit, Michigan, United States

LUKE, S, Catalin Balan G, Panait L, Cioffi-Revilla C, and Paus S (2003)MASON: A Java Multi-Agent Simulation Library, *Proceedings of the Agent 2003Conference*.

LOHSE, G L Biolsi, K. Walker, N. Rueter, H. H. (1994) " AClassification of Visual Representations", VOL 37; NUMBER 12, pages 36,ACM Communications.

MAEDA, J (1999) Design by Numbers, *The MIT Press*.

MINAR N, R Burkhart, C Langton, and M Askenazi (1996) The Swarmsimulation system: a toolkit for building multi-agent simulations. *WorkingPaper 96-06-042*, Santa Fe Institute, Santa Fe.

NESBITT, K V, Friedrich C. (2002) Applying Gestalt Principles toAnimated Visualizations of Network Data. Proc. IV 2002, *IEEE ComputerSociety Press*, London, UK, 10-12 July.

NORTH, M J , N T Collier, and J.R. Vos, (January 2006)"Experiences Creating Three Implementations of the Repast Agent ModelingToolkit," *ACM Transactions on Modeling and Computer Simulation*, Vol. 16, Issue 1, pp. 1-25, ACM, New York, New York, USA.

MACEACHREN, A (1995) How Maps Work, *The Guilford Press*.

NORMAN, D A (2004). Emotional design: why we love (or hate)everyday things. *New York: Basic Books*.

PIPES, A (2004) Foundations of Art and Design, *Laurence KingPublishing*.

RESNICK, M (1994) Turtles, Termites, and Traffic Jams, *MITPress*, Cambridge.

REYNOLDS, W C (July 1987) Flocks, herds and schools: A distributedbehavioral model, Volume 21, Issue 4, Pages: 25-34, Year of Publication:1987.

RUCKERS, R (1993) Artificial Life Lab, Waite Group Press.

SCHELLING, T. (1978). Micromotives and Macrobehavior. New York:Norton.

TISUE, S. & WILENSKY U. (2004). NetLogo: Design andimplementation of a multi-agent modeling environment, *Proceedings of Agent2004*.

TUKEY, J (1977) Exploratory Data Analysis, *Addison-Wesley*.

TUFTE, E (1983) The Visual Display of Quantitative Information, *GraphicsPress*.

TUFTE, E (1996). Visual Explanation, *Graphic Press*,Cheshire, Conn.

TUFTE, E (2006). Beautiful evidence, *Graphic Press*,Cheshire, Conn.

WARE, C (2004). Information Visualization, 2nd Ed., *MorganKaufmann*.

WARE, C, (2002). Purchase H, Colpoys L and McGill M, Cognitivemeasurements of graph aesthetics, *Information Visualization*.

WERTHEIMER, M (1923). Untersuchungen zur Lehre von der Gestalt, II(Laws of organization in perceptual forms). *PsychologischeForschung*, 4, 301-350. Excerpts translated and reprinted in W. D.Ellis (Ed.), *A source book of Gestaltpsychology* (pp. 71-88). New York: Harcourt, Brace and Co., 1939.

WILENSKY, U (2001) Modeling Nature's Emergent Patterns withAgent-Based Languages, Proceedings of Eurologo 2001.

WILENSKY, U. and Reisman, K. (2006). Thinking like a wolf, a sheepor a firefly: Learning biology through constructing and testing computationaltheories. *Cognition & Instruction*, 24(2), 171-209.

Models and Computer Software

ABRAHAMSON, D andWILENSKY, U (2004).NetLogo ProbLab Genetics model.
<http://ccl.northwestern.edu/netlogo/models/ProbLabGenetics>. Center for Connected Learning and Computer-Based Modeling,Northwestern University, Evanston, IL.

AN, G (2004). Innate Immune Response Department of Trauma, CookCounty Hospital.
<http://ccl.northwestern.edu/netlogo/models/community/InnateImmuneResponse>

KORNHAUSER, D (2007). NetLogo Palette Extension
<http://ccl.northwestern.edu/papers/ABMVisualizationGuidelines/palette/>.Center for Connected Learning and Computer-BasedModeling, Northwestern University, Evanston, IL.

SENGUPTA, P & Wilensky, U (2007). NetLogo Series Circuitmodel.
<http://ccl.northwestern.edu/netlogo/models/SeriesCircuit>.Center for Connected Learning and Computer-Based Modeling, NorthwesternUniversity, Evanston, IL.

WILENSKY, U (1999a). NetLogo.
<http://ccl.northwestern.edu/netlogo>. Center for Connected Learning and Computer-Based Modeling.Northwestern University, Evanston, IL.

WILENSKY, U (1999b). NetLogo Models Library.
<http://ccl.northwestern.edu/netlogo/models/>.Northwestern University, Evanston, IL.

WILENSKY, U (2005). NetLogo Disease Solo model.
<http://ccl.northwestern.edu/netlogo/models/DiseaseSolo>.Center for Connected Learning and Computer-Based Modeling, NorthwesternUniversity, Evanston, IL.

WILENSKY, U (2004a). NetLogo DLA model.
<http://ccl.northwestern.edu/netlogo/models/DLA>. Center for Connected Learning and Computer-Based Modeling,Northwestern University, Evanston, IL.

WILENSKY, U (2006). NetLogo Ethnocentrism model.
<http://ccl.northwestern.edu/netlogo/models/Ethnocentrism>.
Center for Connected Learning and Computer-Based Modeling, NorthwesternUniversity, Evanston, IL.

WILENSKY, U (2004b). NetLogo Heatbugs model.
<http://ccl.northwestern.edu/netlogo/models/Heatbugs>.
Center for Connected Learning and Computer-Based Modeling,
Northwestern University, Evanston, IL.

WILENSKY, U (1998a). NetLogo Virus model.
<http://ccl.northwestern.edu/netlogo/models/Virus>. Center for Connected Learning and Computer-

Based Modeling, Northwestern University, Evanston, IL.

WILENSKY, U (1998b). NetLogo Heat Diffusion model.
<http://ccl.northwestern.edu/netlogo/models/HeatDiffusion>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

WILENSKY, U (1998c). NetLogo Rumor Mill model.
<http://ccl.northwestern.edu/netlogo/models/RumorMill>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

WILENSKY, U (1998d). NetLogo Voting model.
<http://ccl.northwestern.edu/netlogo/models/Voting>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

WILENSKY, U (1998e). NetLogo Termites model.
<http://ccl.northwestern.edu/netlogo/models/Termites>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

WILENSKY, U (1998f). NetLogo Flocking model.
<http://ccl.northwestern.edu/netlogo/models/Flocking>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

WILENSKY, U (1998g). NetLogo Segregation model.
<http://ccl.northwestern.edu/netlogo/models/Segregation>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

WILENSKY, U (1998h). NetLogo Decay model.
<http://ccl.northwestern.edu/netlogo/models/Decay>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

WILENSKY, U (1998i). NetLogo Simple Kinetics 1 model.
<http://ccl.northwestern.edu/netlogo/models/SimpleKinetics1>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

[Return to Contents of this issue](#)

© [Copyright Journal of Artificial Societies and Social Simulation, \[2009\]](#)

