DAVID HUER

Practical Wisdom & Profitable Thinking
Innovator - Marketing, Operations, and Disruptive Emerging Technology
T: (778) 828-4933 | E: Huer.David@gmail.com

Websites:	BCIT Technical Web Design:	http://dhuer.htpwebdesign.ca/
	Business Innovation Site:	http://dhuer.htpwebdesign.ca/innoprojects.html
	Design/Community Re-visioning Site:	http://www.behance.net/DaveHuer
	Linkedin:	http://ca.linkedin.com/in/davehuer/

TRULY ORIGINAL THINKING

When I was a boy, I'd spend hours and hours trying to understand what numbers were, because they were holes on a page and when I think about paper, it has thickness of about 2 cm thick. So, I'd inspect all sides of the hole, from underside too, trying to understand what the shape meant. And I learned. Now, I use these skills to solve deep spatial challenges.

When I use pick-up-sticks (to teach innovation thinking), I cognitively walk underneath and among them as if they are scattered tree trunks; as if I am walking through Mexico's Cueva de los Cristales / Cave of the Crystals.

To learn coding at British Columbia Institute of Technology (2013-14), I metacognitively figured out how to bound cloud thoughtspace with a box--then created a Tutorial for other spatial thinkers here: dhuer.htpwebdesign.ca/projects/javascript_jquery_tutorial/tutorial_tut03.html

Teaching a Physically-Challenged Student to Roll a Kayak, Gull River Centre in Minden, 1996: My best whitewater experience ever was to teach a physically-challenged teenager whose arms were twisted how to Eskimo Roll, after national athletes, and other coaches gave up. What I did was become his body to figure out how and where to move it to guide his actions. This involved projecting myself into his body space. http://dhuer.htpwebdesign.ca/innoprojects/inno-minden.html

Pre-Design School

In high school machine shop class, we had an assignment to create salt & pepper shakers using a manually-operated ancient milling machine made to do a max 6 sides. I asked my teacher if I could make a 7sider. He laughed and said it was impossible. After 45 minutes at the machine, I handed him the 7-sided part. I did this by entering "the zone". I had been thinking about the problem for a week, then went to the machine, and held the angular permutations in my head whilst controlling the levers. See: http://dhuer.htpwebdesign.ca/innoprojects/inno-shaker.html

When dating an historic building to 1855-56: after wracking my brain while looking at 8 months of data, and having a niggling thought that the building was from the mid-1850s (not 1858-59), I went to bed one night, and woke up from a dream about all the data cascading back to a specific primary document (year, record type, sub-type) that must be located in the university archives. The archivist showed me an uncatalogued stack of mid-19th C records donated 30 years previously, and we found the document there.

This is what I do. Innovate differently by thinking differently.

CLOUDSPACE THINKING

Many years ago, approximately 1987, I figured out that my practice is to think as a visual “cloudspace" of data points (using "cloud" language back then), by looking at all sides of an object at the same time. When solving a deep problem, I immerse myself deep into the clouds of data, and become/engulf the problem/challenge/ question; quickly compare/sort/combine through many options; and if involving immediate risk and deep challenge, often come to a potential first solution within 2-5 minutes (but then might have to spend the next few months or years explaining it). These “data cascades” are described below.

You can image this as walking into a fog of data. “Path-Dominant Learners” blindly look for a single path because they do not discern the particles that make up the fog, assume there is only one path, and insist that only one path must exist because they know it is there. “Spatial-Dominant Learners” see data fog as a mass of particles and cannot find (or see) the same path every time because the particles are constantly shifting position; so know there is no single path and instead a multitude of paths. It appears that I become the fog, particularly notice all the points, notice the patterns, and then notice the tiny clues that point to interesting stuff. This to me is similar to what Inuit sculptors think about their work: their view is that the subject (animal, being, object) is already inside the stone and they simply free it from the stone. I become the question.

Finding Reference Points

For many years, the challenge with math and other numerically-defined rituals is that I could not see paths in an equation because I get lost in the parts of the equation, get lost in the rich spatial beauty of the shape of the numbers (hence not be able to understand the meaning of the number shapes) and also misplace how math symbols (ex. “=”, “+”) functioned. I didn’t see equations as a system and the numbers and symbols as parts of a system. The challenge with not comprehending symbols as a reference point meant I would "lose" the joining reference point between two halves of an equation. I tried to resolve this by imagining an equal sign ("=") as a bridge between each half of each equation, but then get "lost" again because I would forget the adjoining reference points between each "abutment" and its associated path of numbers and mathematical symbols.

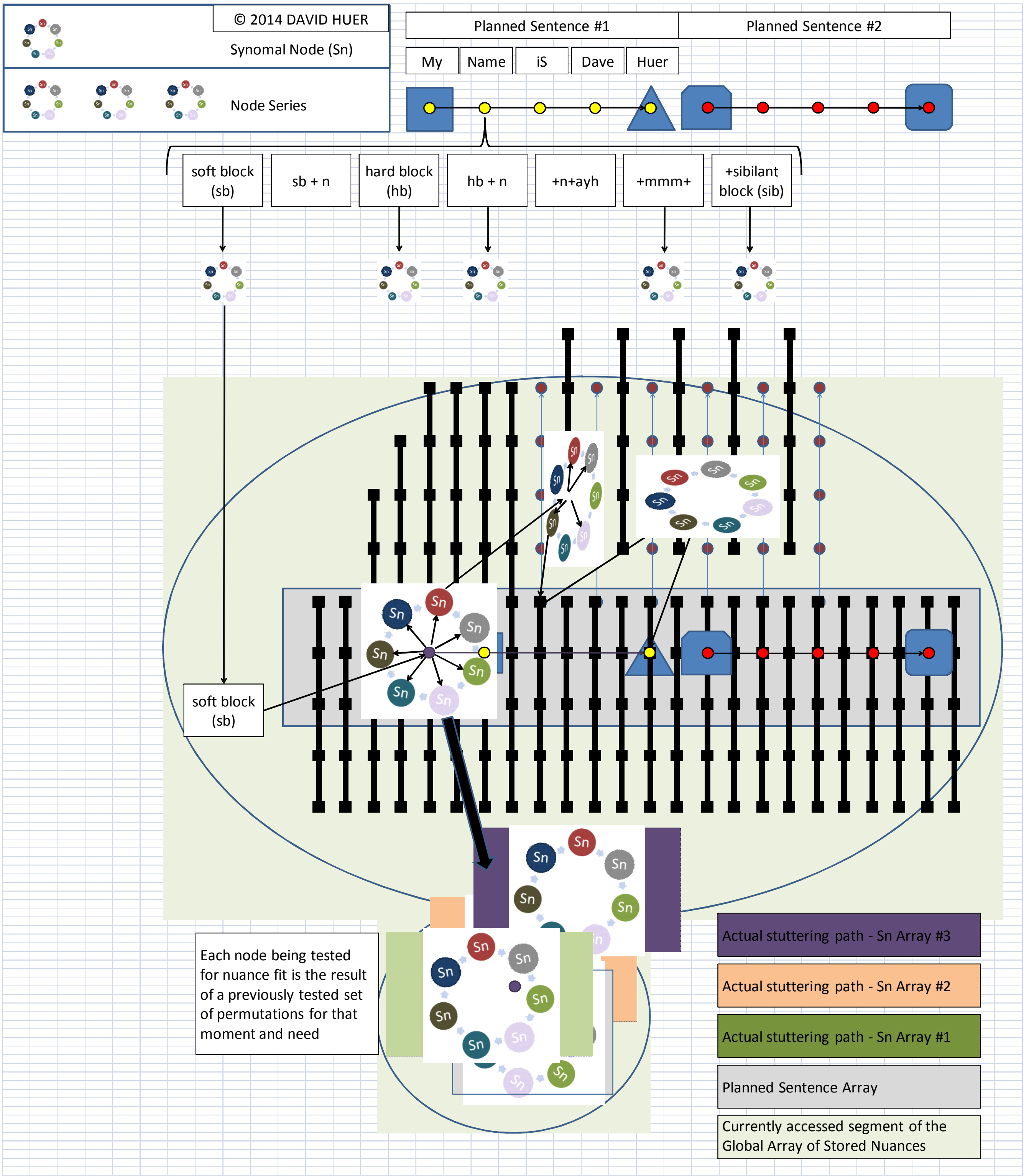
In fact, before inventing cloudboxing, my experience of ritual learning was to see an equation:

> $x + y = z$
>
> as:
>
> "x-symbol" + missing reference point + "plus sign symbol" + "y-symbol" + missing sign reference + "equal sign symbol" + missing reference point + "z-symbol"

It took me innumerable hours to practice for math tests (and often, I “failed” tests designed by linear thinkers for linear thinkers). I have concluded since that because I could not find reference points, I learned and/or taught myself to use the uncomprehended regions where the reference points were supposedly located as:

> (a) sub-bridges between the known parts of an equation, and
> (b) new "end" points for a multitude of entirely new exploratory paths
> (these are Sn nodes, described below)

I’ve had a severe stutter for many years. When I was a boy, the “reference point hunt” challenge helped me get through a stuttering block because I was consciously aware of preparing up to 8 different synonyms arrayed and available in "real time" for every sound nuance I was getting stuck on. This is my **nuance testing model:**



© 2014 DAVID HUER

Synomal Node (Sn)

Node Series

Planned Sentence #1

Planned Sentence #2

My | Name | iS | Dave | Huer

soft block (sb) | sb + n | hard block (hb) | hb + n | +n+ayh | +mmm+ | +sibilant block (sib)

soft block (sb)

Each node being tested for nuance fit is the result of a previously tested set of permutations for that moment and need

Actual stuttering path - Sn Array #3

Actual stuttering path - Sn Array #2

Actual stuttering path - Sn Array #1

Planned Sentence Array

Currently accessed segment of the Global Array of Stored Nuances

For example, a sentence with 5 words intended to be spoken as:

(A + B + C + D + E)

might actually get spoken as a series of what I call synomal clouds of synomal nodes (Sn):

(A1 + B1 + C1 + C2+ C3 + C4 + D1 + D2 + E1)

Synomal nodes happen every time there is a speech block. Every speech block attracts a cloud of nuance sets, the best nuance fits are prepped for testing and each best nuance option is tested for fit. A nuance set ought to be thought of as 3dimensional examination of possible combinations (binary and hybrid meanings). The nuance that fits best becomes the starting point of the new "sentence model" direction, and the entire string is almost parallel to mimic the intent and meaning of the original sentence plan. This happens in milliseconds and is unnoticeable by a non-stuttering observer. You must understand, though: each sentence model is constructed from a global datacloud of words, nuances, and sounds.

You can think of this as a cloud of data points, strung together in a multidimensional array. Data construction to me is multinary, not binary. There is no Yes/No, 0/1. There is instead a balancing act of dataflow. That's where I go. In whitewater kayaking, there are interesting flow structures at the top of a rapid that are synonymous to the event horizon of a black hole. The event horizon of a black hole defines the edge of no return; once past, you're gone into the maelstrom. Similarly, at the head of a rapid, there are times when you can sit at the top of a raging torrent, and not get sucked in, but sit there, languidly, until you choose to enter. It is challenging to go back upstream, though—so there is a balancing place before these event horizons. A balancing place between choices. There are always multiple choices—at the head of a rapid there are at least 6 choices (stay balancing on the event horizon, go downstream, go upstream, go to river left shore, go to river right shore, hop on a rock if there is one).

When ready to solute (create a solution set), I experience what I call a "data cascade". All the data cascades into a series of proposed solutions and often, a system solution; and cascading is sparked by a trigger—a clue that is a key that finally unlocks the puzzle. Cascades are quick events that are noticed as they happen. Similarly, each system model is constructed from a gathered global cloud of data.

Visual/Tonal Mapping

When learning and observing, I am over time able to map the characteristics of the tonal shape of the spaces I move through, and use these visual/sound maps to construct shape and location paths.

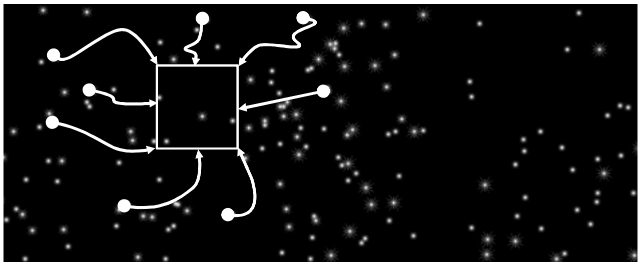
This skill set involves the ability to map, 3-dimensionally locate and store the locations of all the data linkages as a constantly-moving set of data points, and reconstruct the form/shapes around me by approximating the hidden shape from remembered parallel data points. ex. When I walk past a truck and think about the drive train, I deconstruct the approximate pattern of drive trains and "see" the truck as a transparent shape. When I look at a lamp post, I see the visual side and deconstruct it into a shaped column. My caving buddies love the fact that I can accurately draw cave passages from memory.

Take my stuttering explanation, remove the "speech block/Sn response" event pairings, and there is the same method of moving through cloud pattern sets, noticing, cataloguing, sorting the cloudpoints, then coming to a place where a data cascade is triggered. When designing my own projects, it is of course easier to map the project, object, or challenge because I am examining at all sides of it all the time, inside and outside.

Cloudboxing

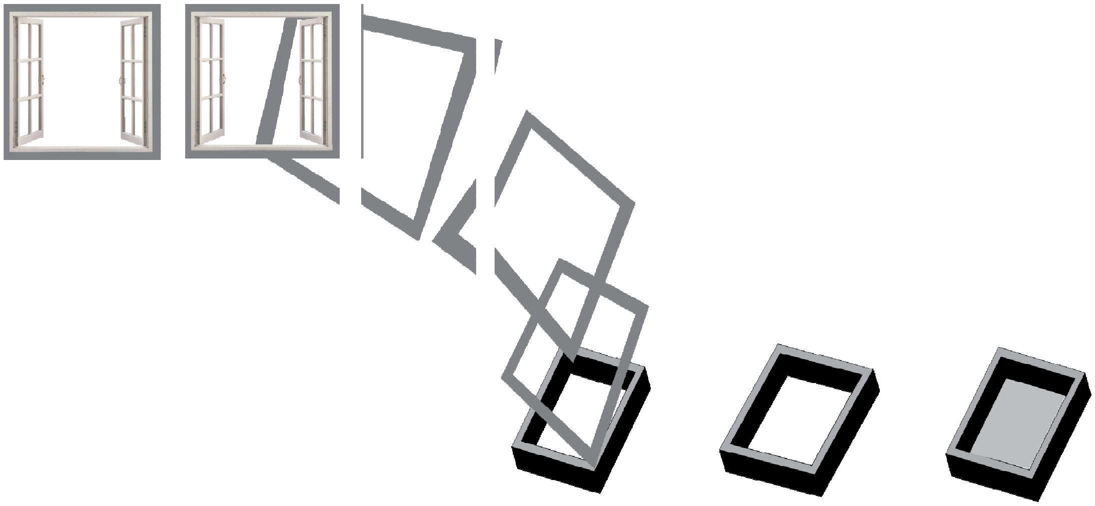
I am able to construct and study complex data sets, but some challenges were beyond me because I could not focus on specific data long enough to study it. High complexity/high remembering requirements—fine. Low complexity/ linear remembering requirements—completely lost. Cloudboxing is exciting because it is a way to solve this ancient locationing challenge.

Simply put, whilst at BC Institute of Technology, I constantly interviewed other classmates and faculty to get a sense of what coding actually is, then taught myself to construct a box out of datapoints. You can picture this similarly to the note about teaching the disabled student how to kayak roll: by gathering a set of datapoints in 3D thoughtspace.



I was able to define the shape as a deep dish pizza box—first by using the desktop browser shape to create an imagined rectangular window shape, rotating that shape to a "bottom" position, extruding 4 vertical walls from it, and then adding a bottom to it. Making the code being worked equivalent to a slice of pizza:

http://dhuer.htpwebdesign.ca/projects/javascript_jquery_tutorial/tutorial_tut03.html



My tutor understood my explanations of what I was doing, and helped me understand that javascript/jquery could be thought of as the "lid" on the box.

This took 4 weeks of intense and painful thinking. As a sense of pain, it is how I imagine left-handed people felt when being forced to learn to be right-handed. Not natural, and painful, but doable.

It doesn't mean I move easily between 2D and 3D thinking. Instead, I translate 2D activity into 3D activity, then do that 3D activity inside the cloudbox. But now I can code, and understand what coders and math people do, and have started to plug jquery arrays into the datastream.