

Site Map: [MAIN](#) / [A Reader's Journal, Vol. 2](#) [Webpage](#) [Printer Ready](#)



A READER'S JOURNAL:

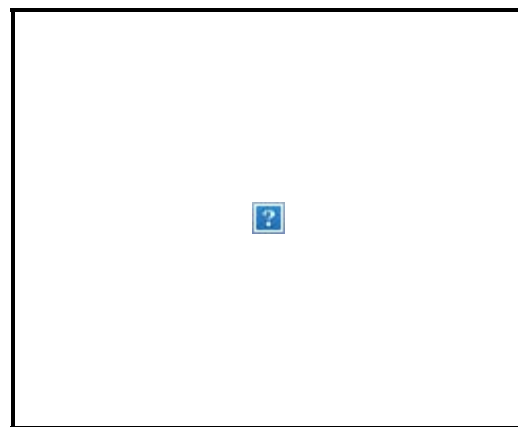
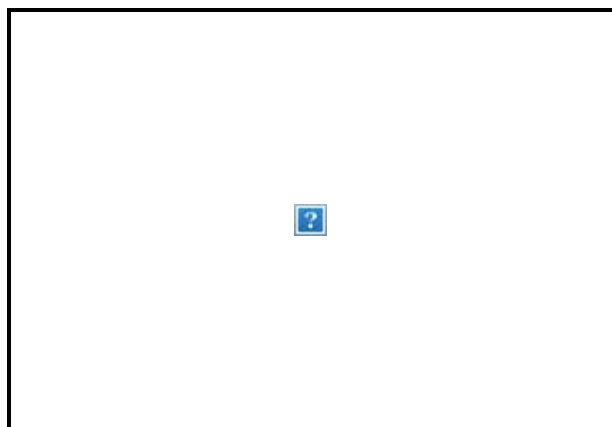
Zen Computer

by

Philip Toshio Sudo

Published by Simon&Schuster in 1999

Book Review by Bobby Matherne ©2007



I write in my books, often in the margins. Sometimes I draw in my books. I use my marginalia when I write my reviews. In this review the two converge: my drawing in the margins of the book appear in the review. These two cute little characters comprise what my friend Ed Manning gave a name to: a date glyph. A date glyph is my personalization of the year, month, and day into a single drawing. These date glyphs adorn and mark the pages of books as I read them. Thus it is possible to track the date on which I read a particular page in a book. Since the expressions on the faces of the figures vary with my mood as I draw them, these date glyphs also mark my mood as I read that particular page. These two date glyphs were drawn on December 29, 2003. The bodies of the two fishes are created by two script m's, one inverted, representing MM or 2000. The "3" is looped into the tail of the right-handed fish. The glyph is also a yin-yang symbol as it incorporates both the male or germinating figure on the left and the female or engendering figure on the right (represented by the eyelashes).

The strangeness of zen and the difficulty of understanding zen are portrayed in the content of the above two cartoons. Philip Toshio Sudo uses the metaphor of the computer to explain zen or the metaphor of zen to explain computers. He calls the book a "Zen Computer" and structures the book as if it were an instruction manual for a 'zen computer' — one which must be booted up during the process of reading the book, as the book is the only incarnation of the zen computer. So read on, boot up, and learn about zen, learn about computing, and learn about yourself.

On the process of booting up a computer, Sudo gives us this brief tutorial:

[page 37] If you are ready to begin the application, it's time to turn on your computer -- in computer jargon, to boot up.

With the initial jolt of electricity, the computer starts preparing itself to get up and running, taking inventory of itself, setting switches, installing programs. In essence, the machine has to kick start itself, "pull itself up by its own bootstraps" before it can start working; hence the term *boot*.

What he says is true, but the *essence* of bootstrapping — while he promises to share it — he simply glosses over it. Here's the essence of the bootstrapping process, as I understand it. Once you understand the process this way, the reason for the expression used to describe it will be illuminated for you.

- 1. To start a computer working you need to load a program into it.**
- 2. To load a program you first need to load a program loader.**
- 3. The program loader can then load programs to run your applications.**
- 4. The program loader is a program.**
- 5. How does that program get in?**

In short: how can you get a program into the computer without a program already inside the computer to load it? It's like lifting yourself by your own bootstraps. Bootstrap Paradox!

In the 1960s our new mini-computers had sixteen switches on the front that corresponded to the 16-bits of two adjacent bytes of computer storage which was its basic unit of storage. When we stored a piece of data into the computer using those switches, we called it "fat-fingering" data into the computer. (This was before we had keyboards and monitor screens for entering data.) To start up a computer back then, we had to fat-finger the bootstrap program into the front panels every time we powered on our mini-computers. Then we could load the program loader and then applications. This process is all automated today using read-only memory (ROM) and hard drives on our present personal computers. ROM holds the bootstrap loader. The boot sector of the first hard drive holds the primitive program loader — which is why your computer will not boot up if that sector is destroyed by a virus or errant program. We may ignore the whole bootstrap process today when all the work takes place behind the scenes of the computer panel, but if we wish to understand how our computers "boot up" we must go behind the scenes and understand the process described by the Hungarian mathematician, János Bolyai, in the quotation that appears at the head of the Chapter "Boot" on page 37:

"Out of nothing I have created a strange new universe."

Each chapter begins with a pertinent quotation such as the one above, and each chapter is filled with stories. Which reminds me of a story. When Gregory Bateson was asked how we would know when computers had reached the level of intelligence of a human being, he answered this way, "We would ask the computer a question and it would answer us, 'That reminds me of a story.'"

In English and Western languages, a "one" is drawn as a vertical line. In Japanese and other Eastern languages, a "one" is drawn as a horizontal line. Since our computers are binary computers, they deal on an internal basis with various combinations of "zero" and "one" or 01. In Japanese, a 01 could be represented as $\underline{0}$ and you will see scattered throughout Sudo's book (e.g. page 154) the "zero" over the horizontal "one". The "one" in English looks like a flagpole; the "one" in Japanese looks like a surfboard. In this next story we can experience how the "one" of the surfboard becomes a "zero" of the balance point when the surfer is on the board and standing calm at the center of the wave of 0.

[page 31] Where do we find this still point of everything that moves? In mind, body, and spirit. A zen lesson designed to illustrate this notion tells of two monks arguing at a flagpole. One says, "the wind is moving." The other counters, "the flag is moving."

The master Bodhidharma, upon hearing the exchange, corrects them both, "Not the wind, not the flag," he says. "Mind is moving."

If mind is moving, Bodhidharma says, we are not calm and at the center.

Once we align ourselves through zen and stand at the center of the circle, we become like the surfer on the wave. Then the still point at the center moves as we move —

moving according to movement, staying ever balanced, always finding equilibrium. As the masters say, the way is known through balancing, not balance.

All these ideas, zen masters express through 0.

Here's another story dished up from the still-point center of the 0 by Sudo:

[page 42] A famous story about buddha tells of a time he gathered his monks for a speech. Wordlessly, he held up a flower. Only one in the crowd, Kasyapa, broke into a smile, Understanding the master's message: the truth transcends words and forms. All things speak of god, be they flowers or stones or boxes of plastic, wire, and microchips. If the buddha held up a computer today, what would you do?

What would you do? I thought as I read the passage that I would stick a horizontal finger at him, representing the "one" which together with the zero form the basis of the computer he was holding up. The one, which when looped together on itself becomes the "zero" which is the other side of the yin-yang of the computer. When we loop on ourselves, as in zazen, which is a sitting zen meditation, we become as one with ourselves and the universe.

The way of *looping* is different in the Western and Eastern cultures, even in something as simple as drawing the figure "zero."

[page 32] In Western cultures, for example, the number 0 is generally written starting at the 12 o'clock position and proceeding counterclockwise back to the 12 o'clock position. In the handwriting of Japan and China, however, 0 is drawn starting at the 6 o'clock position and moving clockwise around to the 6 o'clock position again. This action emphasizes the feeling of arising out of nothing, moving through time, and returning to origin.

We might each echo the words of the female fish in the cartoon glyph above and say to the author of this book: "You zen me." He takes us to the land of the Excluded Middle that transcends the either-or choice of 0 and 1 :

[page 33] The aim of Zen Computer will be to find the intuitive balance between 0 and 1 — not the halfway point at 0.5, but that transcendent place where both $0 \neq 1$ and $0 = 1$, and where neither $0 = 1$ nor $0 \neq 1$.

If this does not boggle your mind, then perhaps the truth of zen is already known to you.

[page 34] As you proceed, you'll find plenty of moments when zen doesn't make sense and yet rings true. Whatever means we use to approach it, that unnameable thing we call zen always has been and always will be; it simply *is*, whatever name we give it. Clap your hands, laugh at a joke. Zen is to be found in every action and every thing — including the computer. The truth of zen is already known to you. All you need to do is realize it.

And, from the next page, our yin-yang glyph speaks to us:



"A watched pot never boils" is an old saying from the West. This speaks to the process of looking at your watch constantly while waiting for some cherished event to occur — the watch never seems to move, just as the pot never seems to boil. Zen masters rarely teach using content, rather, they use process — they do something which illustrates the meaning directly *without* words. Because they operate that way, you learn directly as if you already knew what they were trying to teach you — and, in a way, you do. In a hurry to become a zen master, hmmm, better read this next story, a "parable about an eager young student who

approached a master to learn the art of swordsmanship":

[page 58] "Master," he said, "if I study diligently, how long will it take me to learn the art of swordsmanship?"

"Ten years, perhaps."

"What if I work exceptionally hard? Then how long will it take me?"

"Thirty years."

The student was perplexed. "I'm willing to endure any hardship and make any sacrifice," he said. "I just want to learn the art in the shortest time possible."

"In the case," the master said, "it will probably take seventy years."

Such people are advised to start young — in fact this sort of impatience seems endemic in our youth today. There was a young gardener who was planting his first crop and spent an entire morning carefully placing his sprouts in the ground, covering their roots with soil, and watering them. That night about midnight he awoke in his bed thinking about his sprouts and went out into his garden and tugged on each sprout to help it shoot up. The next morning all his sprouts were dead. Give yourself enough time. And especially give your charges the time they need to develop — provide them a sheltered place to grow, plenty of sunshine, water, and nutrients and then leave them alone. Watch over them, pull out a few weeds, but mostly leave them alone.

[page 59, 60] Align yourself with the true pace of life. If you plant a seed today, don't ask for a garden overnight. A flower grows when it grows. Likewise, when you're installing a new system, preparing a file, or writing a new program, make sure to give yourself enough time.

Then prepare to give more.

This next subject Sudo broaches has to do with obsessions and addictions. He tells of a friend who was addicted to playing computer games and "conquered" his addiction. His friend "decided never to play another computer game." His friend, in my opinion, has not *conquered* his addiction, but merely *replaced* one addiction with another one. Instead of playing computer games all the time, he is now avoiding playing computer games all the time! This process of replacing one addiction with another one is very common in our Western society of vertical-flagpole-one logical reasoning. Apply a little zen and surf that still-point in the center of the wave of zero and you may be able to see that conquering an addiction involves being able to perform some activity other than the none of "zero" or the all of "one". If you have only one way, you're stuck. If you have only two ways, you have a dilemma. Only if you have three or more ways, can you be said to truly have a choice. (Story on pages 61, 62)

Teaching and learning is a lifetime task, rightly understood, not one that just happens when you're young so you can get a job. Get a life, not a job. Sudo places this great quote at the head of his chapter "Learn and teach, teach and learn."

[page 62] *In times of profound change, the learners inherit the earth, while the learned find themselves beautifully equipped to deal with a world that no longer exists.*

— Eric Hoffer (1902-83) Author *The True Believer*

Then he explains how we are "none of us ever exclusively a student or teacher. We are always both at the same time, and must always think of ourselves as both." I wrote this epigram about twenty-five years or so ago to describe that process:

[Thus a Teacher, So Also a Learner](#)

This terse phrase has no verbs so it operates as the still-point of the zero and opens itself to all interpretations, including the one by Sudo in the previous paragraph. "When a teacher is ready, a learner will arrive," is another interpretation often expressed. "When the learner is finished, the teacher will

disappear," is a corollary to that. As an exercise you might work out other ways of interpreting the phrase. I like this way that Sudo talks about the process in his paragraph titled "Show up." :

[page 65] If the teacher is there, you should be, too. Too often we make excuses as to why we can't show up — we're too busy, we don't feel like studying today, we'll go next time. There is no next time. There is only now.

I see this phenomenon in reports I get from some people who read my writing. They'll say, "I browsed your review." or "I scanned your Digest." What puzzles me is that they give a compliment with a wave of the pen and take it away with the next wave of the pen, when they could have as easily given only a compliment. Are they telling me that they don't allow themselves the time to fully experience the world in everything they do, or only that my writing doesn't grab their attention? Sudo says to folks such as these, "Pay Attention".

[page 65] Show up not only in body, but in mind and spirit as well. Many times we're there, but not all there.

Here's another great quote ferreted out by Sudo. (I've taken the liberty to replace the word "men" by "humans".) Below it is the 2004 cartoon glyph it inspired, in which the yin offers a solution, in process, to the yang question posed by Skinner. Note the "4" on the tip of the female side of the glyph.

[page 65] *The real problem is not whether machines think, but whether humans do.*
— B. F. Skinner (1904-90)



[page 83] *All err the more dangerously because each follows a truth. Their mistake lies not in following a falsehood but in not following another truth.*
— Blaise Pascal (1623-62)

This quote by Blaise Pascal reminds us that there are as many truths as paths. That we may err even if we follow a true path if it is not the best path for us. Rightly understood, we have always the option to choose one path or to "blaise" a new path. I hope you follow my line of thought, if not, see the next quote — a poker metaphor. (Note: mathematically, a "line" is an infinity of "points".)

[page 94] *I see your point . . . and raise you a line.*
— Elliot Smorodinsky, Eclipse America Corp.

Many people spend their life perfecting their faults. In the chapter "Drag & Drop" we encounter this next story which helps us to remember to let go of clinging to strictures and rules, to drop something without looking back, and to move on to something else:

[page 96] On a rainy day, the two men came upon a woman looking to cross a muddy road in her silk kimono. "Allow me," said Tanzan. He lifted the woman in his arms and carried her across the divide. Later on that evening, Ekido approached his master. "As monks, we've taken strict ascetic vows," he said. "We're not supposed to touch the opposite sex. How could you pick that woman up?"

"I put the woman down at the side of the road," Tanzan said. "Are you still carrying her?"

This next quotation is from a pioneer in computing, Tony Hoare, which advises us about two ways of software design. In my experience the first way is the far less frequent way.

[page 117] *There are two ways of constructing a software design: one way is to make it so simple that there are no obvious deficiencies, and the other way is to make it so complicated that there are obvious deficiencies. The first method is far more difficult.*

— C. A. R. Hoare, Professor, Oxford University

My favorite chapter is the one on "Programming" because of the seven rules for a tea ceremony. These are the rules I strive to follow when I have guests over to dinner.

[page 124, 125] Japan's most famous tea master, Sen Rikyu, was once asked by a disciple. "What are the most important things to keep in mind during a tea gathering?"

Rikyu replied,

*Make a delicious bowl of tea;
Lay the charcoal so that it heats up the water;
Arrange the flowers as they are in the field;
In summer suggest coolness, in winter, warmth;
Do everything ahead of time;
Prepare for rain;
Give those with whom you find yourself every consideration.*

Sen Rikyu was an extraordinary man. He would, no doubt, agree with this next assessment by Hubbard.

[page 144] *One machine can do the work of fifty ordinary men. No machine can do the work of one extraordinary man.*

— Elbert Hubbard, Author, *A Message to Garcia*

This next quotation by Einstein reminds me of a quote by Jane Roberts, "Where is the tree from which the fruit drops into mind's basket?" Einstein knew that his original thoughts dropped into his mind wordless, from parts unknown, and his job was to create the words and mathematical equations to express the thoughts.

[page 150] *The thoughts did not come in any verbal formulation. I rarely think in words at all. A thought comes, and I may try to express it in words afterward.*

— Albert Einstein (1879-1955)

Then comes one by Will Rogers, the American humorist whose entire life was that of a zen master. He is best known as the man who "never met a man he didn't like." His words below speak to all of us who browse and skim our way through life, always saving time for the next thing to do.

[page 163] *Half our life is spent trying to find something to do with the time we have rushed through life trying to save.*

— Will Rogers (1879-1935)

There are people writing using hyper-text so that their thoughts can be linked directly into the great network of thoughts of the World Wide Web, which is what the www in www.doyletics.com stands for. And yet there are others, the "learned" who, in [Hoffer's words](#), "find themselves beautifully equipped to deal with a world that no longer exists" and print out web pages to read them off-line where the hyper-text is meaningless. (Click on underlined words for examples of hyper-text.) To these "learned sages", reading from the *dead tree* medium is like a religion, and the World Wide Web to them is simply a means for creating pages of paper to read. This will change as book-sized, paper-flexible reading devices with everywhere access to the World Wide Web become inexpensively available. At that time, books will begin to be collected for their iconographical value more than their content value — they will in effect become antiques.

This book is very well written, but there is one passage with a minor problem, which I would like to point out because it illuminates an aspect of the Zen Computer.

[page 195] A good upgrade improves deficiencies and, more important, retains the best of what was.

It seems to me by my reading of what a deficiency is, I don't want the deficiency *improved*, that is, I don't want whatever's wrong with the software made bigger, or better, or more prominent as the definition of *improve* would indicate — on the contrary I want the deficiency *minimized*, or even better, *eliminated!* As for the rest of the sentence "retains the best of what was" — what I find is that upgrades tend to add bells and whistles that I have no need of, all of which makes the software package more cumbersome, harder to use, harder to understand, and more bug-prone than the one I am buying it to replace.

For example, I use a word processor. I want a good, dependable word processor that *only does word processing*. Recently when my package got "upgraded" with a Speech Recognition "feature", suddenly it started crashing in that speech module, even when I was *not* doing speech recognition. The thing that I bought to do word processing better, was suddenly worse than the previous version at doing the one thing I was buying it for. Why upgrade then? Good question. I don't upgrade as a rule. But sometimes operating systems require newer versions of software packages and you are never given the choice of just buying the old version converted to work on the newer operating system. Instead they only offer to sell you the new, improved one — and often I've found it's *only its deficiencies* that were made bigger and the net result was not better at all!

What's the path to Zen Computer? No need to ask — Sudo tells us — because "as the zen masters say, 'You're on it.' " Wherever you go, there you are — on your path. Whether it's your path in your job, in your family, or in your life, there you are — on your path. Use a computer and it becomes part of your life and the results you get from it will be as useful as from any other tool you use — the difficulties you have with it will be idiosyncratic, as if it were designed for you. Ever have the feeling that there's a little guy in the back room of your favorite Chinese restaurant choosing the fortune for the fortune cookie that will be put on your plate — for how else could those words hit so close to home? Using the Zen Computer is similar. It's your path, your computer, and the problems, frustrations, joys, and delights of using the computer will be a reflection of the rest of your life.

