

Models of Illusion

Everyone readily admits that things aren't always what they seem. But are we really applying this knowledge in our daily dealings — are we consciously ferreting out the *illusionary* from the *reality*? I think not.

For instance, although the overwhelming evidence indicates that our important policies are being dictated by self-serving lobbyists — that's not the whole picture.

There is another extremely powerful (but much less visible) agent employed by these parties, that may be their most powerful tool.

The person behind the screen is the computer programmer. And, just like in the *Wizard of OZ*, they do **not** want you to look at this real controller.

I'll probably have to turn in my membership card, but as a computer programmer (and physicist and environmental activist) I'm here to spill the beans about the Wiz.

The first hint of trouble is spelled out in *Wikipedia's* explanation about computer programmers: "The discipline differs from many other technical professions in that programmers generally do not need to be licensed or pass any standardized (or governmentally regulated) certification tests in order to call themselves 'programmers' or even 'software engineers.'" Hmmm.

My layperson explanation is that computer programming is all about making assumptions, and then converting these into mathematical (Boolean) equations.

The big picture question is this: **is it really possible to accurately convert complex real-world situations into one's and zero's?** Hal may think so, but higher processing brains say no. Yet this is continuously attempted, with very limited success. Let's pull the screen back a bit more.

We'll start with an example about how such a model makes assumptions.

One of the computer programs I wrote was for debt collectors. A typical scenario was that a debtor was given a date to make a payment, and the collection company didn't receive it by that time. *What response is then appropriate?*

In such a circumstance the computer program typically makes an automatic contact with the debtor. (Remember there are thousands of these debtors, and it would be prohibitively time consuming for an agency person to manually check into and follow up each case.)

So what to say in this correspondence to the debtor? Well, it comes down to the assumptions made by the computer programmer.

The programmer tries to simplify such situations into mathematical options. In this case they may decide that the question is “does the debtor have the money to make this payment: YES or NO?” This relatively basic choice then leads to a Boolean progression within the program.

How does the programmer (model) decide on YES or NO? Well other indicators would be used (e.g. were prior payments made on time) to come up with a statistical probability.

Of course any computer model is not ONE set of choices, but rather a whole **series** of YES/NO (IF/OR) calculations that lead to a conclusion. In a complex situation (e.g. debt collection, climate change, or financial derivatives) there could easily be a HUNDRED such choices to deal with.

To understand the implications of that, let’s just consider the case where there are TEN such decision points — each with a YES or NO answer. At the end of such a pipeline, that means that there are 2^{10} (i.e. 1024) possible results. **That’s a LOT of different potential conclusions!**

Unfortunately there are actually MANY more possibilities! The assumption that this debtor situation could be condensed down to a YES or NO answer, is not accurate. There are several other real situations that fall outside of YES or NO.

For instance, what if the debtor never got a notice that the amount was due by the date the agency is monitoring? Or what if the debtor sent the money and it got lost in transition? Or what if the debtor made the payment to the original person they owed, rather than the collection agency? Or what if the debtor sent in the money on time, and the collection agency incorrectly didn’t credit the debtor for the payment? Etc., etc.

For the computer program (model) to be accurate, ALL of these scenarios need to be able to be handled properly (legally, timely, etc.). Can you begin to see the complexity here, just with this very simple example of a payment not being received on time?

There is still another significant factor (we’re up to #4 now) not mentioned yet. What about the situation where the debtor hasn’t paid, but it’s because his child has ALS, and he has no insurance? How does a computer programmer write code for more abstract concepts, like “fairness”? In other words, can *ones* and *zeros* be arranged in such a way to represent intangibles? I think not.

So the bottom line question is this: *is there any way that a computer program can correctly handle ALL of these real-world possibilities — even in this simple debt collection case?* The answer is no. **NO!!!**

Another perspective is that we have considerable difficulty just translating the relatively simple thing we call language — e.g. Greek biblical texts into English. *How many versions of the Bible are there? Why isn't there just one?*

Can we possibly hope to translate a process much more complicated than just words? We can certainly try, but clearly the answer is that there is a LOT lost in the *translation* of any complex scenario (debtors, energy performance, etc.) into mathematical equations and computer code.

Some uninformed parties believe that the user has control of all the variables, and can manually (and accurately) change scenarios. That is incorrect, as the user-controlled elements only represent **a small fraction** of the actual number of factors that are built into the computer model.

A similar fallacy is to think something like “we know the assumptions that the programmers made, and are adjusting accordingly.” Wrong!

In writing a computer program of any complexity, there are literally **hundreds** of assumptions made. **The computer programmer does NOT reveal all these to his customer, for much the same reasons that an accountant does not tell his client all of the assumptions made in preparing a tax return.** He goes over a few of the more basic items, and then says “sign here.”

Oh, yes, this example brings up still another MAJOR variable (#7): **the data the programmer uses as the basis for his creation.**

Just like preparing a tax return depends on two parties working together, writing a computer model is a collaboration between scientist and programmer. If the taxpayer gives incomplete or inaccurate data to the accountant, the result will be wrong. *What's disconcerting is that in many cases, neither party will know that the results are in error...*

Similarly if the scientist gives incomplete or inaccurate data to the programmer to use in his creation, the result will likewise be wrong. **AND neither party will know it!**

I hate to keep going on here, but this is important stuff! Believe it or not, there is still one more significant variable (#8) that we have to take into account. After a computer model is generated, there is then an **interpreter** (e.g. IPCC) that translates the “results” for politicians and the public (i.e. the media).

Here's a surprise: these public interpretations are influenced by such factors as political, religious, environmental, financial, and scientific *opinions*. In their public revelations, do the interpreters explain all of their underlying biases? By now you know the answer: **absolutely not.**

When these are introduced into the equation we obviously have strayed so far from scientific fact that it is not even in sight anymore.

Soooo, we need to think VERY CAREFULLY before we take major actions (e.g. spend a **Trillions** of dollars based on climate predictions, wind energy projected performance, etc.) that are almost entirely based on computer models.

What to do? Should we just scrap all computer models?

No, that's the other extreme. Computer models have merit — *but shouldn't be the tail wagging the dog.*

We should realistically see computer models for what they are — tools to assist us in organizing our thoughts, and producers of highly subjective results that are simply starting points for real scientific analysis.

Because of their inherent limitations (which I've just touched on here) ALL computer models should be treated with a very healthy degree of skepticism.

To insure appropriate integrity, ALL computer models regarding matters of importance should be subjected to the rigors of **scientific methodology**.

*If they can't **accurately** and **continuously** replicate the results of real world data, then they should be discarded.* Unfortunately that is not what is happening.

We have gotten so addicted to the illusion that these programs are accurate — and some have become so agenda driven — that we are now adjusting or discarding real world data that doesn't agree with the model. **This is insane!**

If a model has not been proven to fully reflect reality, then it has very limited use, and should be treated with the same degree of consideration that one might give a horoscope.

John Droz, jr.
Physicist and Environmental Advocate
7/22/09 rev a