GEOMETRY

LECTURE 2

DESIGNING MICROWAVE COOKERS

by

PROFESSOR HAROLD THIMBLEBY
Gresham Professor of Geometry

24 October 2002

Designing microwave cookers

Harold Thimbleby

Gresham College, Thursday 24 October, 6pm

Children can use gadgets faster than adults can. What does this say about the design of gadgets?

This lecture looks at the design of microwave cookers. Instead of getting children to help us design better gadgets, we will employ ignorant gnomes to press buttons all day (actually, their mathematical simulation, since getting gnomes to help any design project is too expensive). It turns out that even ignorant gnomes give us interesting insights into design, which would be very valuable to use even when designing for human use!

This second lecture in the series of using mathematics in design will explain and show how this automatic 'gnome testing' of gadgets gets useful design insights.

*

It all started in 1991. Ian Witten (from New Zealand) and I and our kids spent the weekend in a friend's house isolated up in the Rockies. Our friend had gone away and had switched off the electricity, and so one of our first jobs was to get switched on and get the microwave cooker working for supper. The microwave was a model aptly called *The Genius* (Panasonic model number NN-9807).

It took us about 45 minutes to get it working.

I first suspected we might be idiosyncratically bad at using microwaves; neither Ian nor I were expert microwave cooker users. I therefore built a simulation of *The Genius*, and did some experiments on other people. The conclusion was that our difficulty was not unusual. The only advantage we had by being professors was that we didn't immediately blame ourselves for the problem!

The Genius allowed us to get any number between 00:00 and 99:99 on its display. So we naturally thought that the clock accepted 24 hour times. Since it was late in the evening, 22:02 hours to be precise, we tried setting that time. The microwave seized up — you could say it froze. And we had to unplug it to carry on.

We tried 22:05, then 22:15, and later and later times as time went by, until we accidentally set the time to 1:00. The clock then worked!

Having found one way of setting it, we soon realised that we had been misled by the clock. It was secretly a 12 hour clock, willing only to work when set to a time between 1:00 and 12:59, even though the display could be set to any number.

We had had a wager about what the user manual would say. When we eventually found it, we were both wrong: the manual did warn about the 12 hour clock the problem, but in a footnote.

We expected the clock to work one way and it didn't. Indeed, it gave us misleading clues that as it could count so high, it had to be a 24 hour clock. Whilst we were wrong we would *never* be able to set the clock.

*

This story is a simple example of what happens with many systems. When users don't understand how things work they are completely stuck. Many people give up on even such "simple" things as digital clocks. Some people find it is easier to have their clock running an hour slow half the year than to adjust it for summer time.

How could Panasonic have avoided this design problem? Presumably they test their designs on cooks and other potential users: whatever they did hadn't helped them fix the design problem. Indeed, several years later they were still selling microwave cookers with the same design: I saw the manual of one which warned in much bigger writing than *The Genius* manual had that is was a 12 hour clock. Clearly, even getting feedback from user problems isn't enough to help fix a bad design once it has gone into production — except for improving the manual a bit.

Somehow designers need to evaluate their designs before they committed to production. Unfortunately it is rather too easy for designers to be biased when they try to how users will work with their desings. When designing *The Genius* digital clock, the designer probably assumed that everybody uses the 12 hour clock, and they probably tested the microwave on people who shared the assumption. There would have been no design problem to solve because nobody would ever enter

10pm as 22:00, and nobody in the design team or evaluation team would notice this was a potential issue.

Some design errors cannot be found by using people, whether designers or test users.

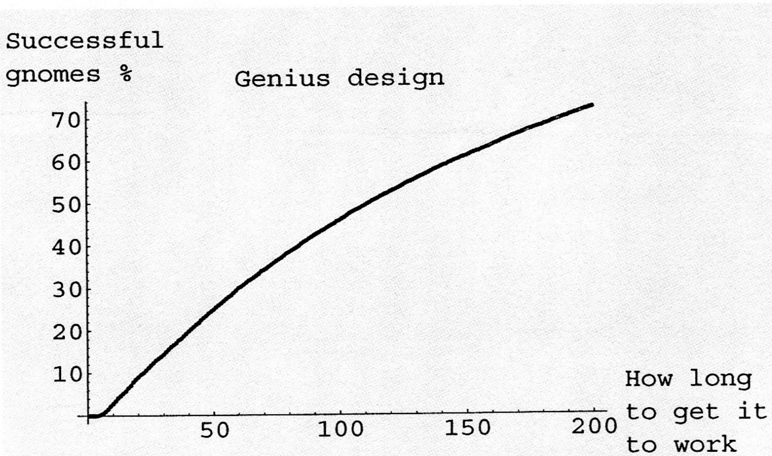
The usual approach to evaluating system designs is to carefully model users, and to try to get realistic models of how they behave. Of course, this is very difficult. In the case in question, we might have some user models, but how would we notice that we'd accidentally set them up to use 12 hour clocks?

It might seem crazy, but a safer approach is to assume nothing about users. Ignorant users therefore might do absolutely anything, and if they behave at random they might assume the clock was 24 hour, or they might even assume it was 100 hour — and discover new design problems nobody has yet thought of. It could be useful to avoid such problems — or at least know they exist, so that palliatives can be worked out.

There probably are no humans who behave truly randomly who could be used this way, but let's use hired gnomes. Gnomes are well known to be very ignorant and know nothing about microwave cookers and aren't worried which buttons they press, or whether pressing some buttons will break them. However, our gnomes are so ignorant about microwaves that they might also try plugging and unplugging the microwave cooker — something Ian and I only found out the hard way as the solution to getting it going when it had frozen up. (If I hadn't told them this, when the microwave froze up, they'd get very frustrated.)

A gnome is really a "random" model of a user and embodies all possible wrong and right ways of using a system. If we have a gnome to test designs, the designer cannot fall into the trap of assuming too much and thereby being misled.

Let's sit a gnome down and get them to try to get *The Genius* to work. We'll count how many steps the gnome takes. Well, obviously sometimes gnomes will strike it lucky, and sometimes they will take ages. So we really need to hire lots of gnomes, and average the results. I hired 10,000 gnomes and sat them down with a computer simulation of *The Genius.* They worked away, and I drew a graph of the results:



The graph shows that almost half the gnomes managed to get the microwave cooker working after 100 button presses, though some took over 200 presses to get it working, and one even took 1295 presses! (Few humans would have the patience without doing more than unplugging the cooker.)

The gnomes seem to find it *really* hard work. Yet if we asked the designer of the clock how many button presses it takes to set it, they might reply just four steps! — to get the clock to work after you've plugged it in, you press [CLOCK] to enter the clock-setting mode, then you press [1-HOUR] , so it shows a valid time (namely, 1 o'clock), then you press [CLOCK] again to start the clock running with that time. Easy, if you know how.

My gnomes take 158, on average, a lot more than 4. This huge discrepancy rather suggests the design could be improved — or at least that the designer's optimism is unrealistic. (We are counting plugging the microwave in as one 'press' — because to get it to work after it has frozen up, it needs unplugging and plugging back in.)

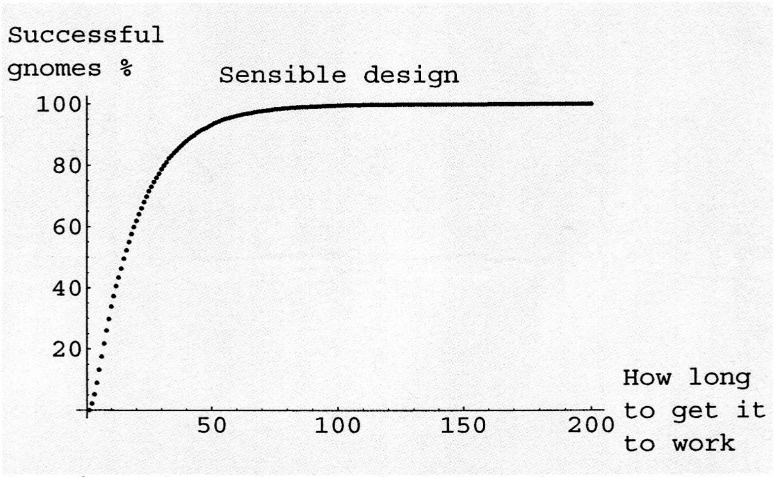
The Genius locks up when it is set to a time outside of 1:00 to 12:59 window. There is absolutely no design reason in principle for a lock up. Removing the lock up (by redesigning it) dramatically helps the gnomes to be faster. If we also change the design so that impossible times, like 27:78, cannot be set, the gnomes get even faster, down to about 20 button presses. Of course, we naturally expect the gnomes to take longer on average than the designer's ideal or a typical human user, because, after all, a gnome doesn't know what they are doing.

Our gnomes have helped us find a faster and easier-to-use design.

Design	Average effort to get it going
Panasonic original *Genius*	158
Debugged not to freeze	107
Sensible design, modified to only show 12 hour times (like a dial clock)	21

So, a little thought — motivated by testing — can get us a design that's on average about seven times easier to use (at least for gnomes if not for humans). But, when we humans use gadgets, much of the time we don't know how they work or what we are supposed to do — and a design change that helps us when we are ignorant and still supports all the original functionality (like cooking chickens) can't be a bad idea. With human users, the faster design has additional the advantage a user would not be able to get the clock to display a "24 hour time" (like 22:02), which was what fooled us, and so would not be misled.

The graph below shows gnomes using the sensible design: about half of them get it working after only 16 presses (in the original design, half took over 112), and no gnome took longer than 191 presses — so even the worst gnome, not just the average gnome, is about seven times faster with the redesign.



The advantage of a random exploration of a design is that it presupposes no specific knowledge of the user (or gnome). This has two advantages.

First, a good designer ought to consider the possible wrong ways in which their design might be used (such as the user thinking it is 24 hour instead of 12 hour, like we did). But there are infinitely many ways of being wrong, and a designer can only think of a limited number of them; a random process, like our gnomes, however, embodies *all* possible wrong ways of using a system. Randomness is a remarkably effective way of testing out designs. After all, if you paid some human users to test out a design they could only test with their own few and fixed preconceptions. Moreover, if their preconceptions were the same as the designers, very little would be discovered about the design that the designer didn't already think they knew.

Quite likely the original *Genius* design was made by a designer who didn't think in 24 hour times, so they never thought to test for them.

So, although a gnomic "random user" is less efficient than a real human user, it cannot be tricked into guessing the designer's tacit assumptions. Gnomes are also a lot cheaper and faster than humans: being cheaper is good ecognomics, and you get them to work faster by using a metrognome (although I used a computer). This ease of testing with 'gnomes' is their second advantage.

It is very interesting that a random gnome can set the microwave clock on average in 50 button presses, whereas I and my colleague took more. Our intelligence was not helping us! We would have worked out what to do faster if we had simply rolled a dice or tossed a coin.

Certainly, being random is a better way of getting things to work than having the wrong ideas. This observation explains why children are so much better than adults at operating gadgets. Since they start off with no preconceptions, they press buttons pretty-much at random. That approach gets results quickly, faster than a systematic approach that an adult would use.

So, we're not old and past it when we can't use gadgets. More likely, we know how they *should* work, but as they don't work that way, we get stuck. Children who don't know how they work (or, equally, don't have preconceptions about 'expensive' errors that scare us adults who paid for the gadgets) get along fine, and quite quickly get examples of how the devices work. Once they have seen the devices work, not only does their confidence grow, but they can also learn from working knowledge.

What's the problem? — put another way

One problem with *The Genius* is that the computer program inside it is childishly simple — which might be another reason why children find it easier to use! It has a bug.

There are four digits in the time display, and simply, each button adjusts a digit. The [10-MINUTE] button increases the tens of minutes; the [1-HOUR] button increases the hours digits. That's all the buttons do. The program is so trivial that [10-MINUTE] always increases the tens of minutes, even from 59 to 69, then to 79. Neither 69 nor 79 minutes are valid times by anyone's clock. The programmer, perhaps pleased with the neat scheme of every button behaving exactly the same, has forgotten that the only button that should work so freely is the [1-MINUTE]. The [1-HOUR] button should not change 2 to 3 if the 10-hour digit is already a 1, because we shouldn't be able to change the time from 12 o'clock to 13 o'clock — because 12 hour clocks don't ever get to show 13. It is absolutely trivial to get the programming right so that "times" out of the 1.00 to 12.59 window simply cannot be set. A modified design would then work like an analogue wrist watch, where you simply can't set the time to something impossible.

Markov models

Gnomes are a fun way of imagining how we could test product designs. In fact, I didn't use gnomes but I wrote a computer program than simulated what ignorant gnomes would have done. Another approach is to use Markov models, which are a mathematical way of exploring what gnomes would do. In fact, once we have a mathematical model of the microwave cooker, it's possible to explore all sorts of creative design ideas. Using *Mathematica*, which is a general purpose mathematician's 'workbench,' the lecture will also show:

- How to simulate the device and get user data
- How to write a user manual that is correct
- How to provide automatic help
- How to find out superfluous features in the design
- Coming up with creative variations in the design
- and how to draw various graphs to get more design insights.

In the lecture, we'll also show ways to halve the difficulty of using a microwave cooker, even if we are as ignorant as gnomes are about microwaves.

Summary

The state of the art in product design is to ignore mathematics, and — sometimes — to over-emphasise human impact. The human impact of a design is of course crucial, but it is difficult to escape misconceptions and cultural assumptions everyone in the design process shares. Only complete gnomic ignorance can provide unbiased insights.

Microwave cookers are easy to understand, and these lecture notes used them only as a simple example. We can see the design issues clearly. More complex devices have the same sorts of design problem, and the consequences of bad design are often much worse than not being able to cook supper.

As we saw with the microwave cooker example, many design faults can be recognised and fixed even while the plans are on the drawing board. There is science to help manufacturers design things properly, they just need us to motivate them to start using it.

References

Harold Thimbleby, "The Frustrations of a Pushbutton World," *1993 Encyclopedia Britannica Yearbook of Science and the Future*, pp202–219, Encyclopedia Britannica Inc., ISBN 0-85229-568-5, 1992.

Harold Thimbleby, P. Cairns & M. Jones, "Usability Analysis with Markov Models," *ACM Transactions on Computer Human Interaction*, **8**(2): 99–132, 2001.

Notes on all lectures can be found at http://www.uclic.ucl.ac.uk/harold/gresham — including the full texts of my references above.