My Research – how my work began

I come from a family with a history of public service, military, nursing, education, local government. Something of that probably rubbed off. In the early 1990s I wrote a large software application then licensed it to a multinational. The annual license fees were more than enough to live off. For a key reason, then I decided to pursue research.

We acquire knowledge by virtue of the operation of our sensory apparatus. The research topic I chose was, well, how could a computer do this in the same sense we do?

It might come as a surprise that the Artificial Intelligence (AI) research community has no idea how this fundamental aspect of human intelligence – perhaps *the* most fundamental aspect of human intelligence – could be achieved in a computer.

So that was my research topic. Try to find out.

I didn't know how the research would go. But I loved research. I loved the process. In a sense the goal was unimportant. I love the challenge. I love learning without instruction. Of finding out how a thing works by examining the thing.

So about three decades ago, I set out. The research was mostly outside of the mainstream, but did include a decade of graduate academic study.

Earlier, in my undergraduate philosophy studies in the 1970s I'd focused on logic and mind. Then after completing a science degree, I'd entered the computer software business.

Over the 1980s I maintained an interest in philosophy of mind. And now in the early 1990s, this naturally expanded to include the philosophical problems of AI.

By the end of 1993, I'd finished my large software application and I was running a small support firm maintaining it at three client sites. We were all starting to relax. It was December. Christmas was almost here.

On 20 December I filed a provisional software patent application for an AI system which I called a cognitive database.

Though when it came time to file the complete specification I was uncertain about the industry applicability criterion for AI at that time, and changed the invention title to data compression, but the algorithms and structures remained the original AI ones. As data compression patents, US 5,748,955 and related US 6,414,610 were referenced by IBM, Oracle, Google, Research in Motion and others. (One would expect the mind to have good data compression.)

With the Christmas break almost upon us in December, work was tailing off. The clients were about to close down for the Christmas holiday period of about three weeks. And as always when I had spare time, I worked on a pet project.

My current pet project was a software system I called an ontological database management system (ODBMS). This system reflected what I'd learned in philosophy studies. The fundamental ontology, a topic of Western philosophy, is a list of the fundamental types of thing. The list is: substance, property and relationship. So if a "thing" exists then it is either substance, a property or a relationship.

Particular units of substance are called tokens. A given token will have values of intrinsic properties such a shape. An intrinsic property is one not easily conceived of as existing separately from the token which bears it. A value of the property of shape is a certain shape.

The Eiffel Tower is a token which has values of various properties: shape, color, size, weight and so on. And it is also a term of various relationships. The terms of a relation are the things related. The tower itself, the metal, the paint, is a term of the relation *is-located-in.* The other term is France, the soil and rock.

This particular relation just mentioned, *per se,* could be expressed x *is-located-in* y. The instance just mentioned is where x is the Eiffel Tower (not the name but the actual metal structure) and y is France, the part of the surface of the earth.

So that's the fundamental ontology I learned about in philosophy studies: substance, property and relation.

My ODBMS was "ontological" because it was based on three sub-types of substance: people, places and (other) things. And on the idea of two worlds: the world of properties, and the world of relationships.

With software generally, programmers are needed to make program changes. For a client, this can be very expensive. But what if users themselves could change a database structure by simply typing in ordinary English sentences (using a limited vocabulary)?

My idea was that the ODBMS would "interpret" these ordinary sentences and make the appropriate database table changes – modify tables, create tables, populate tables – along with adding needed SQL or other program code.

A user might type a sentence like: Joe Blogs is a new customer with phone number <some number> and address <some address>. The name, number and address are properties. I used the concept of the thick ropes hung in sea water for growing mussels. The shellfish grow out sideways. The mussels were the properties. A rope had a unique identifier number.

That was the concept for dealing with properties. But what about relationships? The large software application being used by my three clients was an accounting-warehousing-manufacturing package. In using my ODBMS, another ordinary sentence might be something like: inventory item 465734 is a component of finished good [or sub-assembly] 97856. The issue here is the relation: *is-a-component-of.*

Then I thought, well, there are quite a few different relationships a firm might want to record about day-to-day business. Is it possible to reduce this quantity? I concluded it was – by having base relations and synonyms.

For example, the relation i*s-an-employee-of* (*is-employed-by*), *is-a-component-of* (*comprises*), *is-the-owner-of* (*owns*), can all be reduced to the base relation, *x-possess-y* (*y-is-possessed-by-x*) plus a synonym.

But then I wondered how far this reduction might go. How many base relations are there? And the fewer the better, since a small number reduces complexity of the management system and increases stability and response time. The system need fully process only the base relations.

I estimated about half a dozen base relations. Then Christmas vacation started.

But I wasn't really satisfied with half a dozen. How far could the number really be reduced? What is the minimum?

I was idly wondering about this as the balmy weather settled in as it does around Christmas time. The climate variations of spring had tapered off, replaced with the calmness of the long warm summer days.

At various times one feels happy. One usually knows why. The odd thing was, I started feeling happy but didn't know why. It wasn't a huge happiness, just a background sort of happiness. But it lasted all day. I though, how odd.

Then it happened the next day, too, but slightly stronger. And the next day again. And stronger again. Then the following day, Boxing Day, I was just slothing around doing noting much, just relaxing. There was the quiet, satisfied, solitude one tends to get on Boxing Days.

And as I say, I'd been interested in mysteries related to minds and brains for quite a few years. And I'd concentrated on logic and mind in philosophy studies. Then about lunch time nothing much was happening and the answer just jumped in consciousness.

There was one base relation. There was a single fundamental relationship. All knowledge gained through sense experience is reducible to instances of the relation of temporal contiguity.

I was amazed. The force of the idea was huge. For about three weeks I just swam around in the idea. I didn't want to eat. I didn't want to go out. That was just too trivial. Sensory input seemed irrelevant. It felt like there was a mist over everything. There was no mist, but that was the only way I could describe the experience. The external world became meaningless. The idea was everything. It was the universe. It was perfection. Nothing else mattered.

It answered all the problems of AI. All the problems I'd been thinking about over the years. Not just some of them. *All* of them. They were merely different perspectives on the same simple idea, the same fundamental truth: All knowledge gained through sense experience is reducible to instances of the relation of temporal contiguity.

I'd just lie back on the couch with a huge smile. It lasted three weeks. It was wonderful.

Similar things had happened before and also have since, on different subjects, but not as strong. Starting in the early 2000s, each year I'd drive down the West Coast of the South Island on an annual pilgrimage to the ski fields. As you drive over the Haast Pass and come to Lake Wanaka, the road winds up to the saddle between the lake and Lake Hawea to the north, curving around many spurs.

It's not a smooth road surfaces. It's made of road metal, but well made and flat and consistent, yielding a steady tire noise. I was motoring up to the saddle around the many spurs and enjoying the drive in a slightly powerful car with manual transmission, slowing for a curve, entering the the curve, applying power about half way round, powering out of the curve then slowing for the next spur, then on the power again. It was a rhythm.

Then the tire noise started dropping off. Nothing in the road surface had changed. Nothing in the driving style had changed. But the tire and engine noise started fading away, fading to nothing, until all that was left was the rhythm of the spurs. (Which I thought was odd.)

Almost at the top, a paragraph of ideas bolted in consciousness. The ideas were just there, forcefully. It was a complete understanding of a problem I was working on in the Chinese room argument. It was the problem of how to understand why the things computers internally process are not symbols.

As I reached the saddle, I was stunned. I pulled over in a small parking bay on the left and frantically jotted down notes before the totality of ideas disappeared.

In the late 1990s, I was trying to work out the algorithm of consciousness, or at least what I was calling consciousness, which was basically disambiguation and generalization. I was using a line of Slim Picken's from the movie Dr Strangelove: "doggone it Sweet you told me you'd get me to the primary". But I was using text rather than sound and removing the spaces (to accord with the continuous sound of spoken words).

My question was, how could a system categorize "doggone" as being not about a dog or going anywhere but as a mild expletive, "sweet" (ignoring "ee" verses "ea") as a person's name and not a term of endearment or reference to taste, and "theprimary" as a place? And to do this disambiguation and generalization in a way applicable to any sentence.

For several months I'd been penning structure diagrams trying to figure out the algorithm (the algorithm moved around inside the structures – that was the idea) with no luck at all. So I thought I should take a few days off.

And I was doing nothing much and lazing around and enjoying a bit of relaxation, then a name popped into consciousness. It was the first name and surname of the main character in a book I'd read about 30 years earlier at secondary school. As soon as the name came, I knew the algorithm.

The nature of the algorithm came from the technical nature of the work the character was doing in the book. The algorithm was stunningly simple. People think complicated things are hard to understand. Sometimes the hardest things to understand are the simplest.

Earlier this year, 2022, I was working on a document and thinking that now with climate change starting to cause much destruction, and with the high cost of using human intelligence to mitigate climate change and other global pollution, we need a much cheaper form of human-like intelligence. We need artificial general intelligence (AGI). Our survival and that of other species might depend on it.

The next morning I woke up feeling extremely serene. It was a sublime calmness. This was very unusual. Normally I'd wake up groggily thinking geez! I'd like to go back to sleep!

Later that morning I was driving to the local village for a muesli and coffee, and the scenery appeared as if seeing it for the first time. This was like when you return from a long overseas trip. And I was just thinking how strange this was, when I was jolted by a realization.

There was a major problem with the Turing test. I'd never realized it before and I'd never seen it reported by anyone else. But it seemed important for AI. I was amazed. For a start, I'd been so dumb. Why hadn't I realized it? But on the other hand, why hadn't anyone else?

In the Turing test, a hidden judge asks two contestants, a computer and a human, questions by text alone. According to Turing, the computer takes the part of a man using a teleprinter. In other words, the computer is a robot. It has (one or more) eyes and fingers in order to use the teleprinter the human contestant was using.

That's how Turing describes the test. But in the test as performed, the computer contestant is wired directly into the judge's keyboard. It has no eyes or fingers. It can't see the judge's questions. It can't see anything.

The human contestant sees the questions. They print out on the teleprinter's paper roll (or nowadays, a screen). The human sees the text and understands the meanings of its shapes. But all the computer gets is what comes down the wire. And the text of the judge's questions doesn't come down the wire. There is no text inside the wire. So how does the computer know the questions?

Answer: it doesn't. But if so, why does the judge think both contestants do see then answer the questions, when one contestant never sees them? This is the problem: the problem of direct wiring.

On analysis, it reveals so much. I thought surely someone has reported it, so I hunted through all the (many) books I have on Turing and on the Turing test – nothing. Then I searched online. Again – nothing.

Now back in 1993, a large multinational corporation was using my software and I had an independent income from the annual license fees.

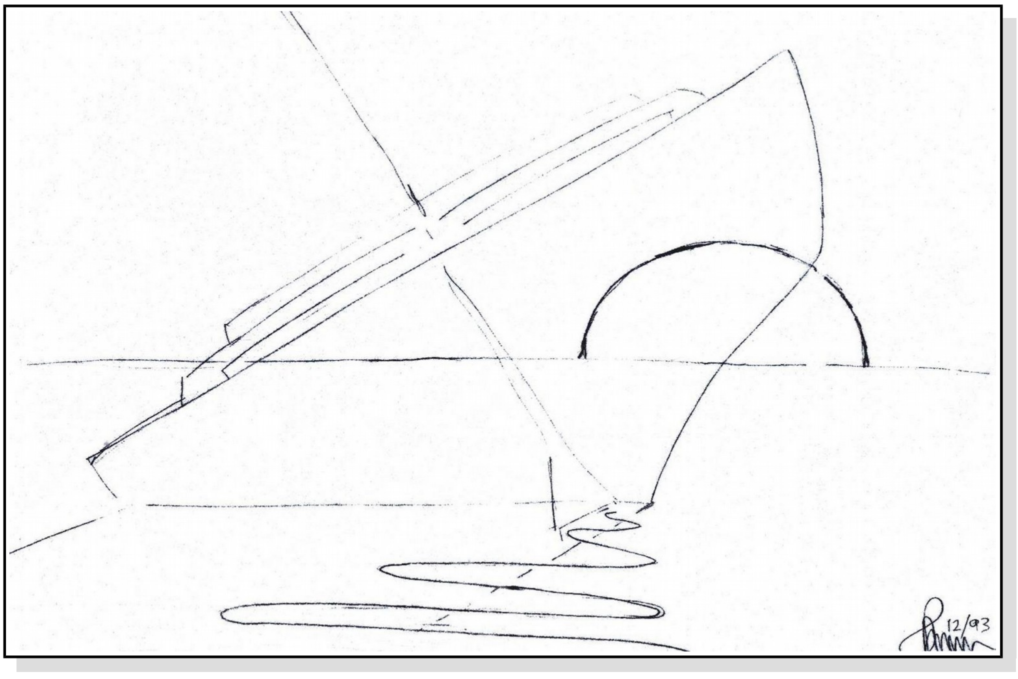
I assumed the Boxing Day idea was the same sort of thing some ascetics get who live in caves and fret about some religious contradiction for decades. A wafer and flesh: one and the same? You must be kidding. An entity being one thing and three things at the same time. Impossible.

I told Jack Copeland about my Boxing Day idea of temporal contiguity and how I got it, and he said: You can't say that! People will call you an <a certain surname goes in here>(!). Which I thought was quite an over-reaction.

It was clear to me that the temporal principle might be wrong. But on the other hand what if it was right? The way it came to me was very forceful. That must count for something. It came by way of some mode of getting fundamental answers outside of the usual process of deductive reasoning, premisses, evidence, conclusions.

But the difficulty was, the idea came with no premisses. I couldn't see how to justify it. I couldn't see how to explain why it might be right. And it seemed so counter-intuitive. But what if it *was* right? Why not (given an independent income) try to find out?

So that's the reason I set out on a three-decades-long journey. I'd studied Art at secondary school and liked drawing. I did a drawing in the last week of December 1993, which expressed how I felt. It seems to have a strange combination of elements: serenity, journey, ego, transparency of reality.



Since the end of 1993 I've been trying to develop the new concepts needed to explain the principle of temporal contiguity.

Over the pandemic, ideas have crystallized a bit to the extent I think I can put up a reasonable case for a new paradigm, a new conceptual framework, for understanding knowledge and how a computer (so-called) could realize it.

I've found this process of developing new ideas very hard. But in the recent solitude have managed to get to a point where it now all makes integrated sense. I'm confident I've got an answer. I'm not sure it will make sense to anyone else.

Brief summary of basic findings

- Computation cannot explain intelligence. Intelligence is explained by togetherness in time. Happily, computers (so-called) can react in the needed ways to togetherness in time.
- The atoms of knowledge in motion are instances of the relation of temporal contiguity in sensory streams as they enter the system proper.
- The instances can be recorded by a computer as instances of knowledge at rest.
- In knowledge at rest, temporal contiguity in the stream is recorded with pointers. This makes permanent the relation in time as the units of the stream enter the system proper.
- Knowledge in motion is created by the sensory process of transduction.
- The relation of temporal contiguity within the stream mirrors temporal contiguity of the elements of the environment detected by the sensor.
- Thus something in the outside world does survive transduction, namely instances of the relationship of temporal contiguity.
- Also repetition of instances of temporal contiguity in the streams mirrors repetition in the environment detected by the sensor. This allows separation within and between the streams of accidental contiguity verses casual contiguity.
- Certain structures of stored temporal contiguity avoid noise and incomplete input.
- Certain structures of stored temporal contiguity avoid combinatorial explosion.
- Certain structures of stored temporal contiguity begin with the most general then, as learning continues, enlarge to become less general, but the more general elements are still present and available.
- Sensory input is continually activating stored experience including past experience of the effects of motor action, and respective structures are continually primed to react to the next atoms and compounds of current input on the basis of past experience.
- The stored instances of the relation of temporal contiguity *per se* plus the values of properties of the units related, is what allows the inner structures to know and understand the outer world.
- Inner structures made of records of temporal contiguity within and between the sensory streams have semantic content and intentionality.

---end---