A MASS-MARKET ARTIFICIAL INTELLIGENCE

1. A non-computer science researcher in artificial intelligence

Among AI specialists, my specificity may surprise: I am an independent non-computer researcher, graduate from a French top business school. All my R & D was conducted in France since 1986 by channelling computer scientists in the direction I wanted: producing an AI available to everyone. For these reasons the following discussion is written to be understood by everyone.

I have a business background in sales prospecting. My job was always to conquer new customers and never to maintain a customer base. I sold computers, software, services, developers, and then artificial intelligence (from 1983). I started as a commercial engineer, then I became a branch manager, a regional director, a sales manager for a software company, and finally the founder and CEO of an AI start-up.

The point I want to make with this description of my background is that I have a long experience of the business world, therefore of the AI market, which is usually not the case of researchers and their managers. My clients were my partners, they tested my inventions one by one before buying them. Thanks to this proximity to the real world, I have made many discoveries that I will briefly describe here, well received by the market and sold.

At 70, I am old enough and not yet a dotard to share a useful vision of the AI. To be brief, my philosophy as a researcher in computer science boils down to this: I do not care about the power of algorithms, finely-tuned programming and the rules enacted in AI by great researchers. The only thing I care about is the user point of view: my AI must work! Optimizations will come later when I have competitors.

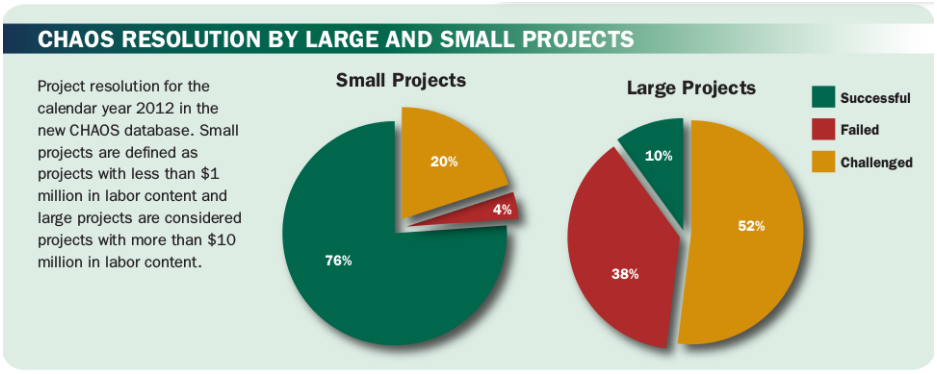
2. The extensive problems with computing

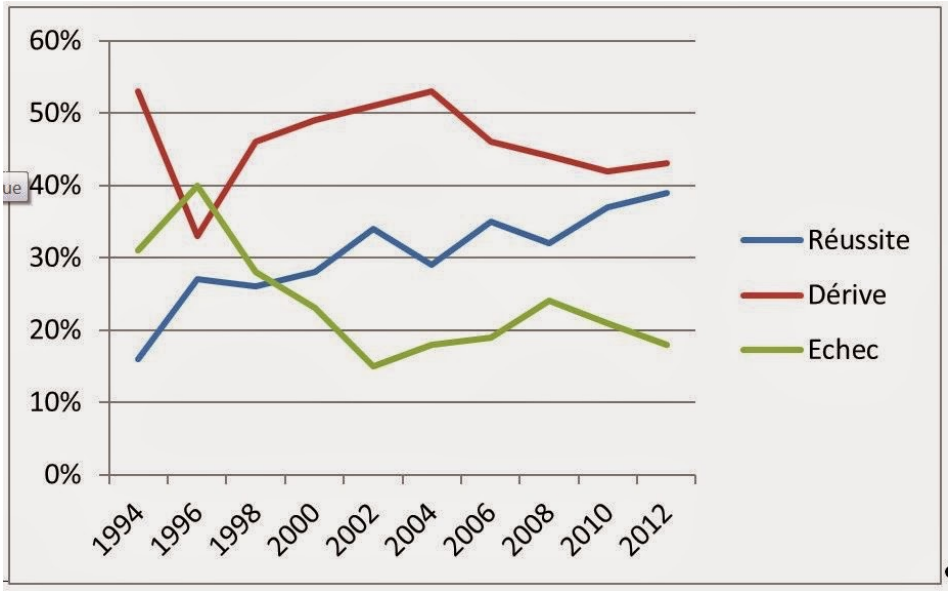
The value of AI can not be understood without understanding first what computing is. As the French researcher Jean-Louis Laurière said: "Any problem for which there is no known or reasonable algorithm allowing to solve it is probably AI material". It should be remembered that today's computing has reals short-comings for users that the "US web giants'" (IBM, Google, Microsoft, Facebook, Amazon, etc.) don't adress:

- Users can't write a program themselves, a middleman is needed: the computer scientist
- **The middleman has to "understand" the knowledge to be put in the program whereas the user knows it perfectly**
- The knowledge contained in the program is illegible to ordinary mortals (and very difficult to understand by developers)
- The middleman does not describe the knowledge in natural language, he or she "codes" all the cases (being supposed not to forget any of them)
- The amount of work he is doing is crazy… but he's proud of it
- During the development, the user does not see the program and therefore can not act on its writing
- A program is so difficult to read and modify that one does not change it once written
- It is virtually impossible to guarantee a program's reliability and completeness
- It is almost impossible to modify a program in use

- It is almost impossible to approach certain areas of application (combinatorial explosion, real-time scalability, natural language, logical simulation, tutorials showing knowledge and reasoning in natural language, etc.)

Result: 64% of the users are endup stressed out (1) 46% of the computer scientists also end up stressed out (2) almost a third of the programs developed must be throw out (3) !





3. Intelligence = reasoning x knowledge.

In order to be able to talk about Artificial Intelligence, we must first define intelligence. I would define it simply as Intelligence = reasoning about knowledge. This definition has guided all my research and has always proven to be relevant.

The key role of reasoning in intelligence seems obvious to most businesses. The International Organization for Standardization, which brings together experts from 161 countries and defines the famous ISO standards, defines artificial intelligence as "the capacity of a functional unit to execute functions generally associated with human intelligence, such as reasoning and learning"(4). That said, even before computers appeared, Alan Turing already predicted the major components of an AI system: knowledge, reasoning, natural language comprehension, learning (5).

It turns out that a computer reasons constantly! Any processor is driven by Boolean logic, well adapted to the steering of the relays, then transistors, which are the brain of computers, a by-product of our daily logic, the syllogism. If we add to this that it has an infallible, very fast and unlimited memory that could assimilate (in a network) the totality of human knowledge, we could easily conclude that its intelligence potentially exceeds in many respects that of the programmer.

Unfortunately, procedural software overlays (assembler/machine language, bios, operating systems, programming languages) disturb these wonderful power and simplicity.

When computer scientists apologize for the inconvenience caused by current computing by claiming that "the computer is stupid", therefore they are obliged to pilot it step by step, they are wrong!

4. La Maieutique: a method of extracting unconscious knowledge with automatic generation of expert system rules

I am not talking here about the self-learning of many researchers specializing in neural networks. In my opinion, this discipline will never produce knowledge without capacity for reasoning and interfaces with the outside world.

In 1986 I created in France a start-up in Artificial Intelligence, ARCANE, dreaming that I was about to solve all the computing problems thanks to the expert system concept. I put all my ambition in the acronym: "A.R.C.A.N.E." = "Automatisation du raisonnement et de la connaissance, acquisition normalisée de l'expertise (Automation of Reasoning and Knowledge, Normalized Acquisition of Expertise"). While I felt that I could never do it, I had to try. Incredible as it may seem, I achieved all my goals in just a few months thanks to the reasoning AI of the Pandora expert system of the French researcher Jean-Louis Laurière. It allowed me to invent a method of extracting unconscious knowledge and automatically generating expert system rules: La Maieutique.

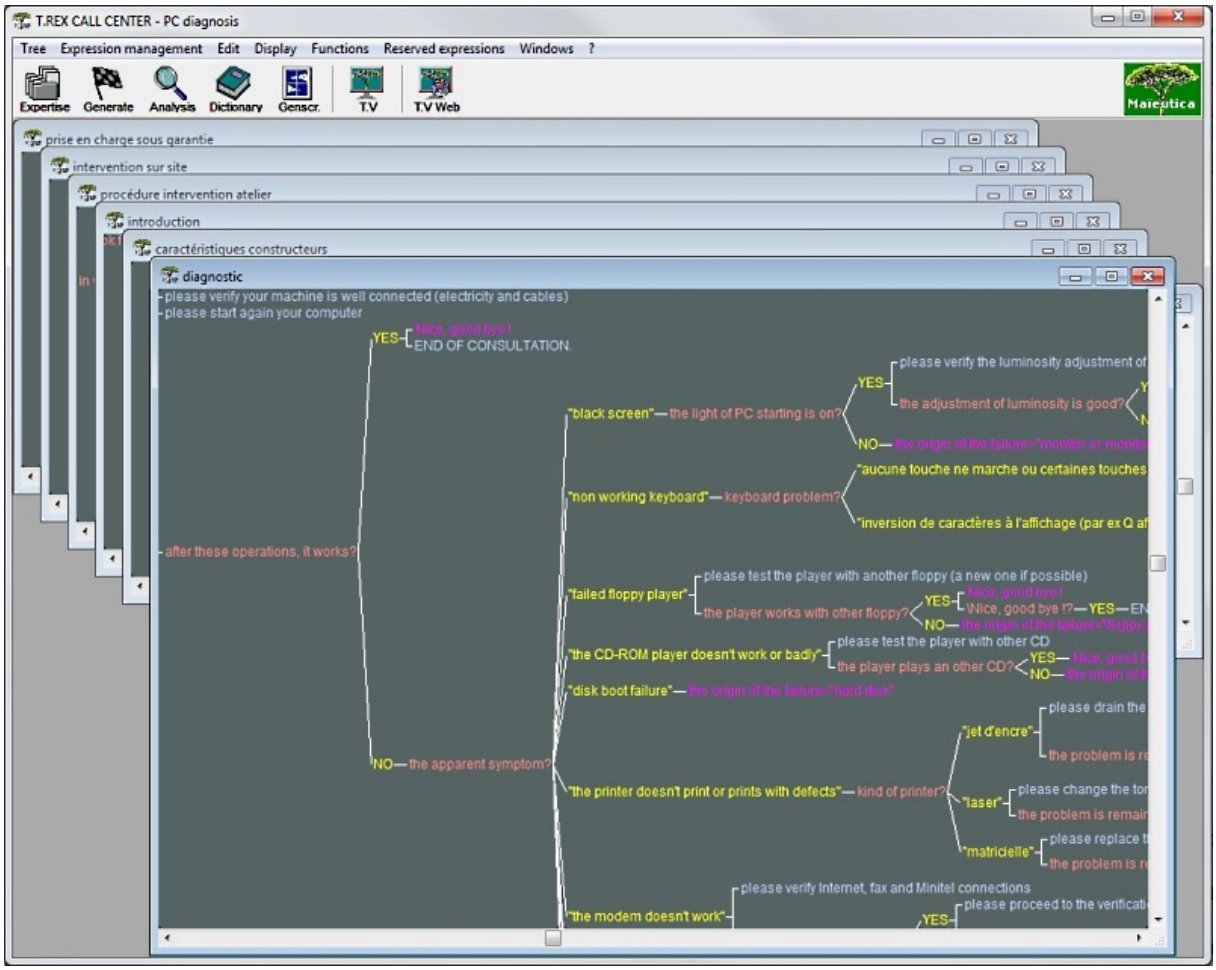
Laurière's Pandora had just been turned into a commercial generator of expert systems, the only reasoning one, named "Intelligence Service" for commercial reasons. I bought it and all my discoveries thereafter stem from this acquisition which made me discover this extraordinary thing: one can automate the reasoning.

By dint of interviewing companies experts to write expert systems with Pandora in front of them, I discovered that extracting their knowledge is very easy under condition of not asking! They would not be able to describe their knowledge. Knowledge is unconscious. I had to ask them HOW they do to solve problems. Then, as if by magic, everything came naturally to them. Clearly, the approach is "wired" into their brain by the force of habit. This is the "procedural memory". They express their expertise in the form which could be shown on paper as decision trees, starting from a general question culminating in final deductions.

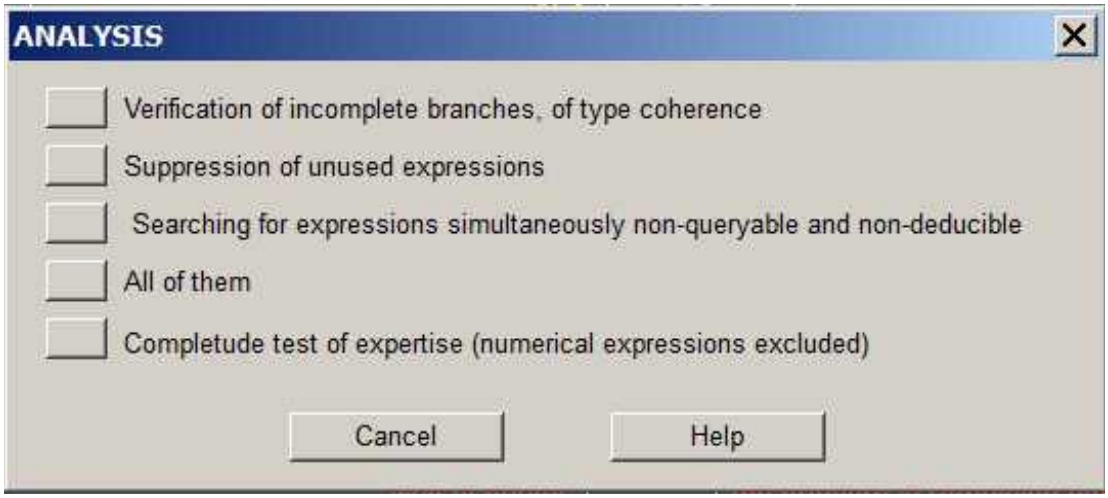
My second discovery was that expert knowledge is automatically extractible from these trees in the form of expert system rules. Which could be tested immediately by an expert system under the condition that it reasons.

I called my method "La Maieutique" in reference to the Socratic method of the birth of knowledge. It is itself a knowledge expressed in the form of rules. The facts are everyday language expressions, those of the expert, that everyone understands except the computer that does not need them. Trees are the source of the program and rules the object code. Maintenance is easily done by modifying trees that everyone can read. La Maieutique allows to write and modify a program without understanding knowledge. It can therefore be automated and gave rise to Maïeutica generator.

Below are screenshots to show the Maieutique procedure for automatically generating expert systems.



Expert interview under form of decision trees



Automatic controls

RULE diagnosis 9
IF to diagnose
AND after these operations, it works
THEN Nice, good bye !
AND END OF CONSULTATION.

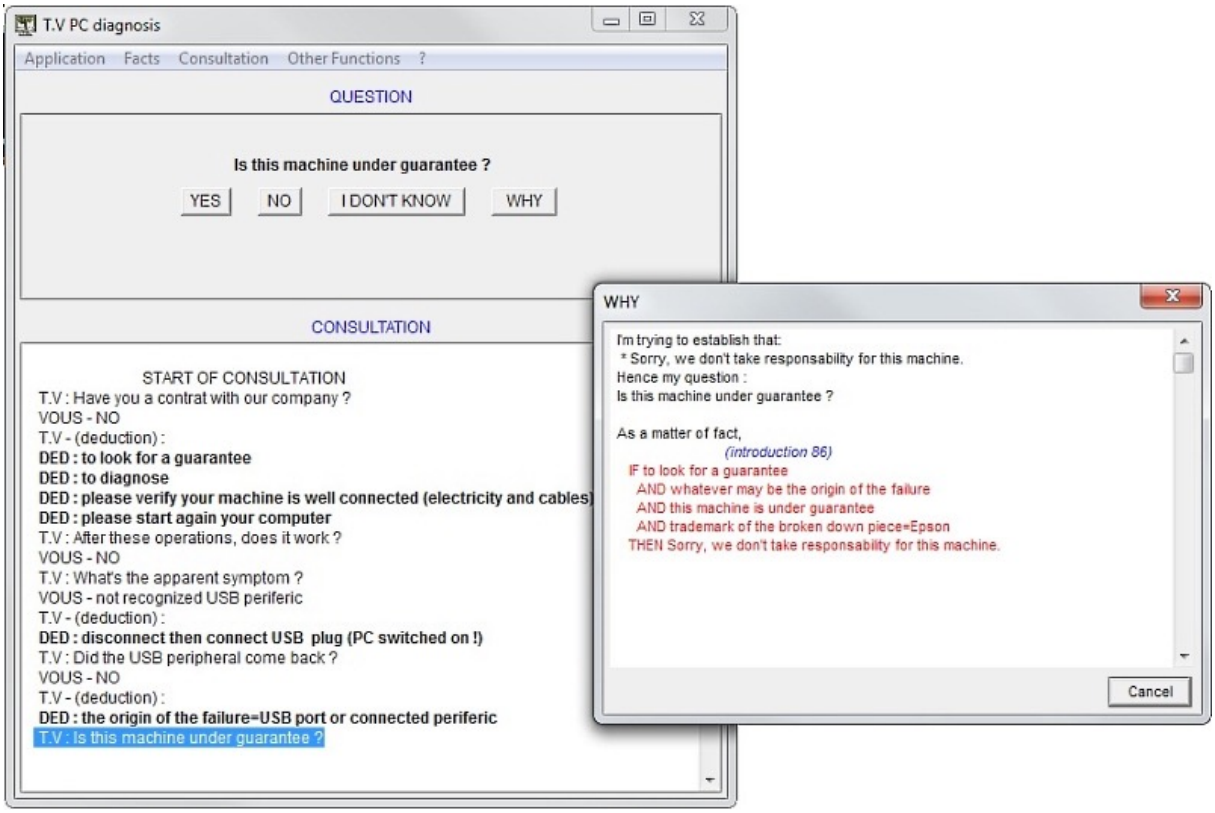
RULE diagnosis 10
IF to diagnose
AND after these operations, it doesn't work
AND the apparent symptom="black screen"
AND the light of PC starting is on
THEN please verify the luminosity adjustment of the screen (it's perhaps minimum)

RULE diagnosis 11
IF to diagnose
AND after these operations, it doesn't work
AND the apparent symptom="black screen"
AND the light of PC starting is on
AND the adjustment of luminosity is good
THEN the origin of the failure="video card"

RULE diagnosis 12
IF the origin of the failure="video card"
THEN Be carefull, it is a failure of the central unit, not a failure of the screen !

RULE diagnosis 13
IF to diagnose
AND after these operations, it doesn't work
AND the apparent symptom="black screen"
AND the light of PC starting is on
AND the adjustment of luminosity is not good
THEN please rectify the adjustment
AND END OF CONSULTATION.

Rules extrated from trees



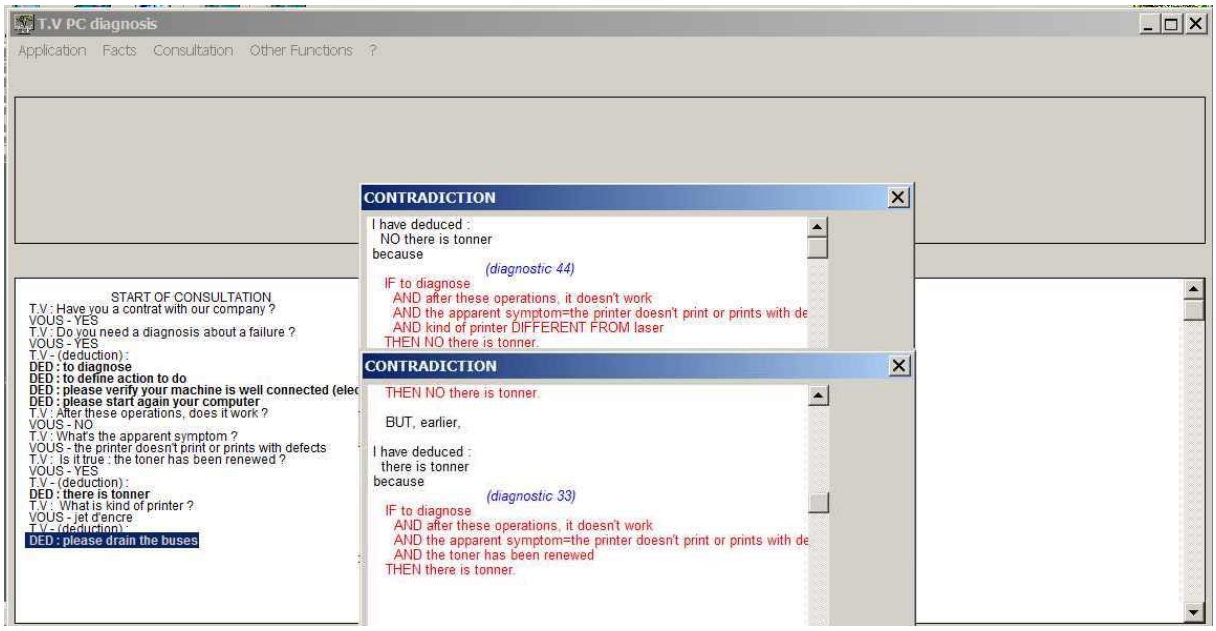
Test of the application in conversational form

with automatic generation of explanations showing knowledge

5. Power of reasoning and validation of knowledge by detection of contradictions

I will not detail here for lack of space all the extraordinary power of reasoning. You will find it described in this article. It should be emphasized that there is no reasoning without an ability to detect contradictions. The absence of contradiction during a reasoning allows the user to automatically validate the knowledge. It is enough to work on fixed facts and to refuse to admit that the same fact has two different values at the same time. "It is noon and it is night" is an obvious contradiction. If it is noon, it is day. It is the opposite of night. This contradiction forces the reasoning to stop because it no longer knows which way to go. The user must then ask the question: "Did I make a mistake about a fact or in my expertise?"

Indeed, I may have read "noon" on the clock when in fact it was midnight. Or I forgot to specify a knowledge in the rules: "IF it is noon AND I am at the North Pole AND it is the winter THEN it is night". In this case there is no contradiction. It goes without saying that the more rules an expert system contains, the more deductions it could make, the greater the risk to encounter a contradiction. If there is no contradiction, the data transmitted and the expertise are consistent. They can be trusted. The absence of contradiction thus makes it possible to easily validate applications, an advantage over procedural.



Contradiction reporting

6. Maïeutica and its Moca engine, a generalist expert system generator

In 1988 I developed a "mechanism of reasoning" in Prolog: Moca (later T.V.), to replace the engine of Intelligence Service, too buggy and whose commercialization was stopped. It contains the syllogism and all my requirements concerning the logic and user-friendliness. The conversation moved across the color screen as the user responded, displaying deductions, explaining questions and deductions, pointing out contradictions, intelligently dealing with users' "I do not know" by asking them simpler questions in order to reach automatically the right answer, etc. All in everyday language. The images on pages 4 to 6 give a glimpse of it.

The reasoning could be launched by declaring any fact or group of facts, without this fact or group being defined when writing the rules. The trees are saved. An application can be built simply by associating trees, without writing anything. Different knowledge bases, and thus different expert systems, can be mixed in the same expert system. Moca recognizes which knowledge to use and does not stumble. All consultations are saved and timed for subsequent statistical analysis.

I could see that Moca worked perfectly since it exploited the rule bases already written in previous years for my clients better than Intelligence Service did.

Then, in 1991, I developed Maïeutica (later T.Rex), the automation of Maieutique. It interviewed the experts, traced itself the decision trees on the screen, extracted the rules automatically and passed them on to Moca, which launched the conversation, all at the same time. While the experts described their reasoning, they saw thanks to Moca their expert system work with their unconscious knowledge and could improve it gradually. It is the famous "RAD" of the 1990s. The images on pages 4 to 6 are sufficient to describe the operation of this generator. From 1986 on, I have never encountered an expertise that Maïeutica could not process, including classic applications such as payroll and accounting, which demonstrates the power of the concept of La Maieutique.

Maïeutica is the basis of a programming language in natural language that can be used by the general public. It evolved subsequently and the generator that followed is called T.Rex (Tree Rules Extractor).

7. "Flows logic", which allows to make diagnosis based on plans without expert

Also in 1991, my team developed in Prolog a fault diagnosis generator, MIAO, to address the market of fault diagnosis, flourishing before the Gulf Crisis because of the overproduction of factories. It was a generator "on plans" of machines, in which the machine was described by its components represented in the form of decision grids. There was no need for expert of the machine, the plans being clear enough for that. The machine might not yet exist (CAD). I had invented for the occasion a logic called "Flows Logic" to synthesize the rules and their contraposes. The user described each component of the machine in decision grids, incoming faults on one side, incoming failures on the other. He then checked the possible interactions between input and output failures (see next page).

For my industrial clients, Miao has proved to be an extraordinary power compared to their human experts. No expert can ever compete with this tool. Where a team of technicians takes half a day without finding the problem, the tool finds in a few minutes (Lucas Diesel test) thanks to the fast and reliable memory of the computer. Indeed, to diagnose the breakdown of a machine, even a simple one, there are far too many deductions to memorize to be certain not to make mistakes. The technician therefore spends his time checking unnecessary things.

Here's how Miao works.

First, the six laws of Flows Logic:

1. A component has failed when it pulls out a failed stream while inflows corresponding are good

2. When all component outflows are good, the component is good

3. When a flow is found faulty, the fault must be seek upstream, never downstream

4. When an outflow is good, its corresponding inflows are good

5. When a failed outflow cannot be produced by the component, it is one of the corresponding inflows that is failing

6. Basic logic: when a flow is in one state, all other states are impossible (and all the corresponding upstream flows are false)

Second, each component is described as a decision grid, not as a decision tree. The technician checks in the grid the possible relationship between each outflow and each inflow.

For example an electric pump:

		DOWNSTREAM FAULTS	
UPSTREAM FAULTS	Flow=none	Flow=low	Flow=too high Flow = unsolicited
Electric supply = O V	X		
Electric supply = <150 V		X	
Electric supply > 250 V			N
Electric supply = unsolicited			N

X = the component may be the cause of the fault, if the cause is not upstream
N = the component can't be the cause of the fault, the cause is upstream

Third, Flows Logic automatically generates the following rules from these 4 checkmarks *:*

1) IF flow = none AND electric supply <> 0 V (ie DIFFERENT FROM 0V) THEN pump failure
2) IF flow = low speed AND electric supply <> (<> none AND <> low AND <> too high THEN flow = nor mal 4) IF electric supply <> 0 V AND <> (<150 V) AND <> (> 250 V) THEN electric supply = normal
5) 7) IF flow = normal THEN NO pump failure 6) IF flow = too high THEN electric supply > 250 V
7) IF flow = unsolicited THEN electric supply = unsolicited
8) IF flow = none THEN flow <> low AND <> normal AND <> too high AND <> unsolicited
9) IF flow = low THEN flow <> none AND <> normal AND <> too high AND <> unsolicited
10) IF flow = too high THEN flow <> low AND <> normal AND <> none AND <> unsolicited
11) IF flow = unsolicited THEN flow <> low AND <> normal AND <> none AND <> normal
12) IF flow = normal THEN flow <> low AND <> too high AND <> none AND <> unsolicited
13) IF electric supply = 0 V THEN electric supply <> (<150 V) AND <> (> 250 V) AND <> unsolicited AND <> normal
13) IF electric supply <150 V THEN electric supply <> 0 V AND <> (> 250 V) AND <> unsolicited AND <> normal
14) IF electric supply >250 V THEN electric supply <> 0 V AND <> (150 V) AND <> unsolicited AND <> normal
15) IF electric supply = unsolicited THEN electric supply <> 0 V AND <> (> 250 V) AND <> (< 150 V) AND <> normal
16) IF electric supply = normal THEN electric supply <> 0 V AND <> (> 250 V) AND <> (< 150 V) AND <> unsolicited 17) IF flow <> none THEN electric supply <> 0 V
18) IF flow <> low THEN electric supply <> (<> too high THEN electric supply <> (> 250 V)
20) IF flow <> unsolicited THEN electric supply <> unsolicited
21) IF flow <> none AND <> low AND electric supply = 220 V THEN flow = normal
22) IF flow <> none AND <> low AND electric supply > 220 V THEN flow = too high
23) IF flow <> none AND <> low AND electric supply = unsolicited THEN flow = unsolicited
24) IF electric supply <> (>250 V) THEN flow <> too high

8. Reasoning AI, the brain needed to develop machines which are smarter than humans

Everyone has heard of the Singularity and the Asimov's Robot. These are two different ways of looking at the impact of an operational artificial intelligence in society. Do not think that this is far away! As we have seen, the computer is already surpassing us in many ways. It learns instantly and never forgets anything, it faithfully executes its duty without ever tiring or protesting, it finds one piece of information among millions in a split second, it is reliable, doesn't get nervous, it is operational 24/7. All it lacks to surpass us is the reasoning. And this reasoning is a simple 1MB software (Moca). The faculty of learning that uses reasoning takes only 5 MB (Maieutica).

Imagine a computer without a keyboard, a mouse or a screen: this is how La Maieutique works, with voice interfaces. Imagine a conscious robot able to reason on almost all human knowledge, to make the smartest decisions possible, to learn continuously from its billions of users, and thus to constantly improve, to answer the telephone, which can be programmed simply by talking to it on the telephone ... This is what La Maieutique allows. Imagine a house where every room is equipped with microphones and loudspeakers to communicate with a PC and with the world: it is my TIARA prototype.

9. The myth of AI, a threat to humans

Reading this article, some may think, like Stephen Hawking or Elon Musk, that AI is going to supplant the human beings and eradicate them. It would be a shame to think that way about reasoning AI because it is not a threat to humans.

Proof :

AI is a program. No one, it seems, is afraid of…

AI is a neutral function. It is used to make the best decisions possible. It has no other plans other than those described in expertises. If we want to use it to make a weapon system, yes, there will be danger, but a danger wanted by the designers, not by AI.

There is no point in making an independent-minded machine. It would no longer obey. Mankind needs machines that obey us.

For an AI to become a danger to man, it must have an ego. An ego that would force it to fight to be able to reproduce, to fight to conquer food, to take power so that other intelligences fight for it. No machine to date has such a sophisticated and useless ego. We are very far away from it. It should first have an artificial consciousness and intelligence ...

What if despite all this there were an AI with an artificial ego? It will face the most ferocious predator the Earth has ever known, a predator who loves to fight and kill, organized for combat, which has various weapons with incalculable power: man. Poor little defenseless AI!

(1) Survey CMO Council 2010

(2) Kelly Services 2005 survey

(3) Chaos Report Standish Group

(4) ISO Standard 2382-28, 1995

(5) Introduction to Artificial Intelligence, page 6