

LA MAIEUTIQUE

Sommaire

<u>Sommaire.....</u>	<u>1</u>
<u>Candidate Name and Location.....</u>	<u>2</u>
<u>Short presentation of my “singularity”.....</u>	<u>2</u>
<u>Genesis of my discoveries: from self learning to conscient machine.....</u>	<u>4</u>
<u>La Maieutique: self learning producing natural language programming</u>	<u>5</u>
<u>Conversationnel: an essentially human skill.....</u>	<u>9</u>
<u>Miao and Flow Logic : the power of reasoning in fault diagnosis.....</u>	<u>10</u>
<u>Tiara and voice T.Rex: the intelligent machine.....</u>	<u>12</u>
<u>Intelligent machine - conscient machine.....</u>	<u>13</u>
<u>Prospects.....</u>	<u>14</u>
<u>Examples of Conversationnels developed with Maieutica/T.Rex and Miao (by experts and a collaborator of Mr. de Lespinay, without computer specialist).....</u>	<u>15</u>
<u>Proposed Citation</u>	<u>16</u>
<u>References (apart from list of articles in newspapers).....</u>	<u>17</u>
<u>Michel Le Seac’h.....</u>	<u>17</u>
<u>Steven Morvan.....</u>	<u>19</u>
<u>Jean-Paul Baquiast.....</u>	<u>21</u>
<u>Bertrand Tavernier</u>	<u>22</u>
<u>Jean-Paul Souris.....</u>	<u>23</u>
<u>Dominique Bayle.....</u>	<u>25</u>
<u>Jean-Bernard Wilhelm</u>	<u>26</u>

Candidate Name and Location

Name (first, middle initial, last): Jean-Philippe de Lespinay

Address: Château de la Mouhée

City: Chantonnay

State:

Zip or Postal Code: 85110

Country: France

Telephone: (33) 682 92 69 03

Fax: (33) 251 42 50 11

Email Address: jpl@tree-logic.com

Short presentation of my “singularity”

- 1.1. I invented a simple and intuitive method called « La Maieutique » (Maieutics) in tribute of the Socratic method, which solves two ever present problems in computer science:
 - automatic extraction of experts unconscious knowledge in everyday language
 - visual representation of know how and human knowledge, legible for all people.

- 1.2. I demonstrate expert reasons on certain facts, never uncertain, using his procedural memory. This is why any expert thinks and decides quickly without applying calculations on his reasoning. « He may be sick » must be managed like a certain affirmation. The human interprets the "maybe" of the computer in his own way depending on the context. As in any conversation between two people.

- 1.3. There is a strong logic to admit that any software is an expert system. It always contains 1) an expert knowledge to be automated and 2) the reasoning of the developer who implemented this knowledge. As with la Maieutique an expert system is very easy to write and even easier to maintain by anyone, it is obvious that any program should be developed with it. A tool

like T.Rex solves most problems posed by traditional programming: expert knowledge extraction, completeness, bugs, reliability, slow programming, difficult to maintain, unreadability, no explanation, unable to deal with application domains highly complex, reasoning, combinatorial or scalable.

- 1.4. I solved the problem of natural language programming for both classic programs and expert systems with an universal black box reasoning engine, « Moca », totally independant from knowledge domain, using natural language knowledge extracted by La Maieutique.
- 1.5. I developed a tool set designed for use by experts themselves or any interviewer, « T.Rex » (Rules EXtractor by Trees). This software includes La Maieutique methodology and Moca. T.Rex interviews the expert and represents his know how as decision trees. As his answers progresses he sees the program running and evolving. Thanks to a conversational interface he can test and modify the program at any time. Nobody ever touches the rules. La Maïeutique manages them to avoid human errors. There is no limit to the number of rules (500, 1,000, 10,000) therefore no limit to the size of the program. The first expert system written with la Maieutique, "Josephine" (1986), had more than 1,000 rules and was developed in 3 months by two experts and an interviewer. Since 1986 I have tested numerous fields of application. This method works regardless of the field of knowledge as long as we have an expert. It fails only when the expert is not an expert, that is to say, he/she has not built a method of resolution.
- 1.6. There is an evidence that a tool like T.Rex is also the solution to the Turing test as it develops programs which dialogue with users naturally, explain their reasoning in natural language, show and teach their knowledge, detect contradictions, simulate cases without asking unnecessary questions, know to react intelligently to users « I don't know » answers.
- 1.7. For the specific case of troubleshooting, I developed the « Flow Logic » associated with decision grids in an expert systems generator called « Miao ». Here, no need for experts: the knowledge for diagnosis lies in the diagrams of the machine and a logic that wants we go upstream the defective flow to find its origin. As you will see below, Miao has an extraordinary powerful logic.
- 1.8. I also demonstrated that a reasoning computer could give the illusion of human intelligence and consciousness by encapsulating a voice T.Rex in a vocal chatterbot (for PC): « Tiara ». Such a computer, or robot, equipped with a microphone and a speaker has no need for mouse, keyboard, or even screen, like humans... It converses with users, learns their knowledge,

teaches it to others, memorizes their guidelines, constantly adapts to their desires, shares what he knows with other computers or robots. It is an association between two intelligences, man and machine. Such a computer is constantly improving like a child prodigy. As time goes on, it will eventually seem smarter and more conscient that man thanks to its reasoning into a huge and inalterable memory.

Genesis of my discoveries: from self learning to conscient machine

First, a scoop: I'm not an IT specialist. I can't program and I knew almost nothing about AI when I made my first discovery in 1986. I graduated from a French high business school ([Euromed](#) 1971). I worked all my life as a commercial prospector in the computer field (since 1973). I've been selling AI solutions since 1983. I became a private researcher pushed by my market. My inventions in AI were guided by the need, ie customer's orders, big companies.

Searching my name on the web will return lots of information about me, albeit in French. In English, you'll find [here](#) my biography and [there](#) a page about my first invention, la Maieutique (Wikipedia article projects). In 2011 I wrote most of the Wikipedia [Expert System](#) article with AI history, its strengths and weaknesses, somehow the continuation of my French article about the disappointing state of art of AI published in [Science et Vie](#) 20 years before (1991). [www.tree-logic.com](#) website features my technology and various little conversational expert systems (in French, sorry!).

In summary, I have shown since 1986 by various implementations in companies that AI is a reality and not a hazy project finally abandoned in the 1990s. Look at this [list of press articles](#). It is actually possible to give an intelligence comparable to humans to a machine by giving it only two things: self-learning and reasoning. For the rest, computer has basically multiple extraordinary faculties alien to man: it never forgets anything, it is never wrong (once bugs are resolved...), it "thinks" at supersonic speed, its memory is virtually unlimited and infinitely clonable, consequently it is virtually immortal. Through self-learning, a computer absorbs human knowledge thousand times faster than a human being and never forgets it. It becomes mankind's eternal memory accessible to everybody. Through self-learning it *writes programs* much faster and better than a team of IT specialists. Because programming and learning are the same things. Through reasoning, a computer is able to think, make decisions, act instantaneously without indecision in the interest of its user, converse intelligently with him, simulate his inventions in its mind, test validity of his ideas and projects before they are built, detect contradictions, explain, diagnose, advise, teach its knowledge, *execute programs* it wrote thanks to self learning, become *conscient*. With both, you guess, we have no more a machine, but a sort of genius which becomes ever more genial with time, knows everything and takes decisions with an intelligence that

transcends us, most valuable partner that humanity has ever had, an eternal partner. Like a god child...

In 1986 I founded my first company, ARCANE, to advance R&D in AI. I developed a self-learning method, "la Maieutique" to automatically write expert system rules in front of experts, and I developed the first version of a reasoning engine in 1988: "Moca". I sold them to various big companies. In 1991, I automated la Maieutique in a program called Maieutica containing Moca. It was the first reasoning expert systems generator. I sold hundreds of copies. Around 1995, by dint of developing expert systems in all domains with Maieutica, I became convinced that *any program is an expert system* since it contains expert(s) knowledge and its role is to execute this knowledge. It would benefit greatly to be replaced by a knowledge base and a reasoning engine. I modified Maieutica in order it could develop any program, conventional or not, pluggable into conventional applications (including as dll): T.Rex (Tree Rules EXtractor). A computer scientist friend, Jean-Bernard Willhelm, confirmed me in a striking way. He tried to sell my technology to computer companies with high software engineering arguments (in vain...) and we are partners since then.

The resistance of computer services against my AI was too strong for my taste. I could not make enough money. I decided to drop businesses and address the public, in principle fascinated by intelligent machines. In 1999 I founded a start-up with venture capital, TREE LOGIC, to develop "Tiara", an intelligent and vocal chatterbox able to program for its users (for PC). First step of my clever and conscious computer. A PC that average users can program simply dialoging with it, even dialoging by phone. Tiara was a success. After two valuation reports by Ernst & Young company I was awarded the coveted FCPI label, a state label given to very innovative French companies in order to facilitate the raising of capital dedicated to innovation.

From 1986 to 2000 I marked the spirits in France by my IA as shown in the list of press articles. I showed that a machine could think, converse and even give signs of human consciousness (see following chapters).

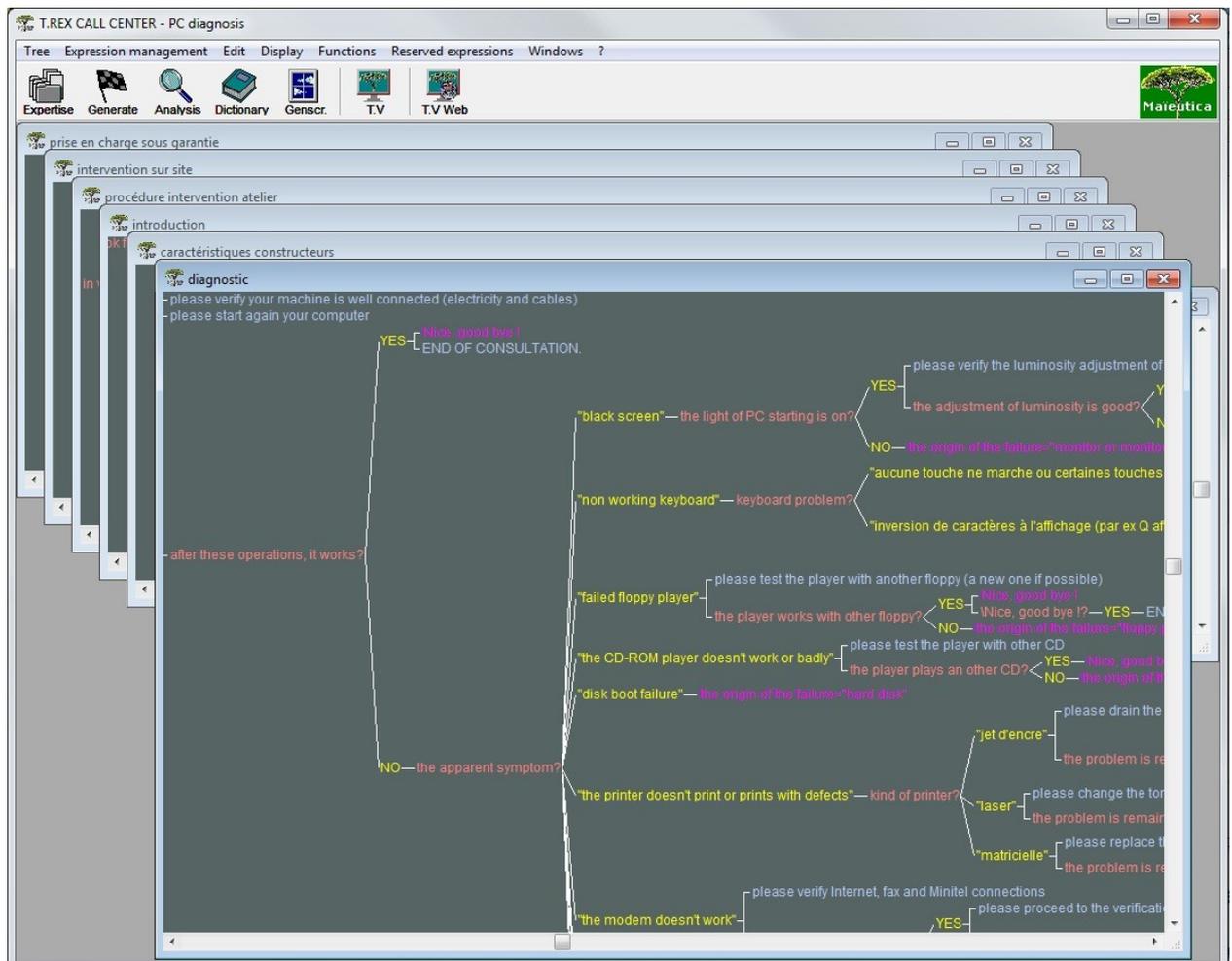
La Maieutique: self learning producing natural language programming

To understand the scope of my inventions you must know their history. My first discovery, in 1986, was a salesperson's discovery. I absolutely wanted to sell expert systems to businesses because I was absolutely convinced that AI is the market of the future. I choose an excellent "shell": Intelligence Service, an unknown French expert system generator having the first engine reasoning of history. In fact it was, without any modification, the expert system Pandora designed by Jean-Louis Laurière in 1982, an academic researcher who got nothing but contempt for this great achievement and always refused to comment. Pandora used zeroth order logic, the

only AI logic worthy of human logic in my opinion, I stress this.

I discovered first it's not possible to sell an expert system to an IT department, they are not ready to listen. You must sell it directly to company experts. When two "normal" people (me and the expert) talk free of computing pollution they eventually understand each other. After writing laboriously rules of expert systems for Intelligence Service during a few months I realized that experts are unconscious of their knowledge, that everyone knows in computer science... But in fact it is present hidden in their procedural memory. A memory called "know how", "experience" or "expertise". Asked about their know how they perfectly describe it by decision trees in natural language because procedural memory has a tree structure.

I programmed La Maieutique in a software called Maieutica and later T.Rex. Here is how procedural memory appears in this software:



It just so happens that it is very easy to extract rules from decision trees:

```
PC diagnosis.rxt - Bloc-notes
Fichier Edition Format Affichage ?

@ REGLE diagnostic 14
diagnostic 14
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"

@ REGLE diagnostic 15
diagnostic 15
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 !

@ REGLE diagnostic 16
diagnostic 16
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.

@ REGLE diagnostic 17
diagnostic 17
IF to diagnose
AND after these operations, it doesn't work
AND the apparent symptom="black screen"
AND the light of PC starting is off
THEN the origin of the failure="monitor or monitor cable"

@ REGLE diagnostic 18
diagnostic 18
IF to diagnose
AND after these operations, it doesn't work
AND the apparent symptom="non working keyboard"
AND keyboard problem="aucune touche ne fonctionne ou certaines touches ne fonctionnent pas"
AND there is one light or more lighted on the keyboard
THEN the origin of the failure="keyboard"

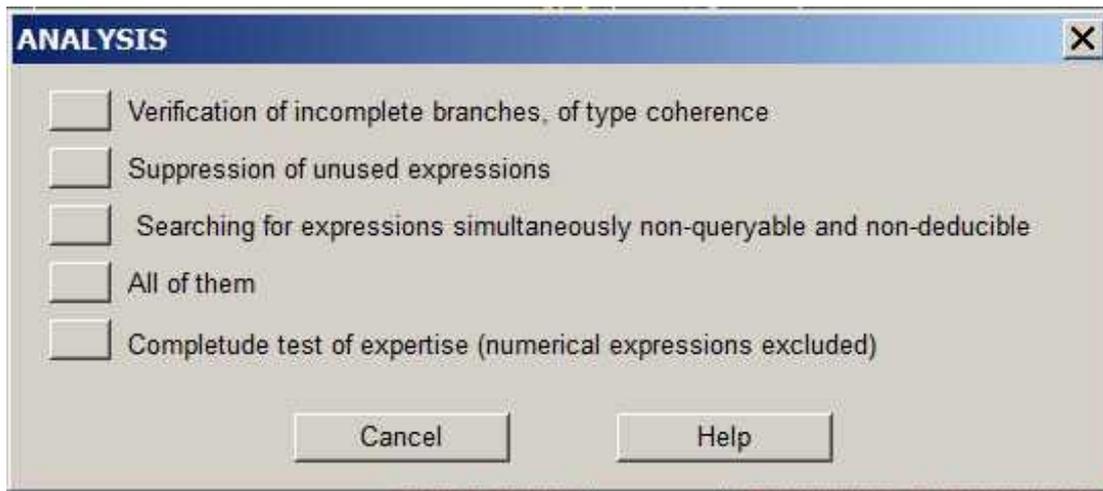
@ REGLE diagnostic 19
diagnostic 19
IF to diagnose
AND after these operations, it doesn't work
AND the apparent symptom="non working keyboard"
AND keyboard problem="aucune touche ne marche ou certaines touches ne fonctionnent pas"
AND there is not one light or more lighted on the keyboard
THEN the origin of the failure="mother-card"

@ REGLE diagnostic 20
diagnostic 20
IF to diagnose
AND after these operations, it doesn't work
AND the apparent symptom="non working keyboard"
AND keyboard problem="inversion de caractères à l'affichage (par ex Q affiché à la place de Q)"
AND your OS="windows"
THEN region parameters verification

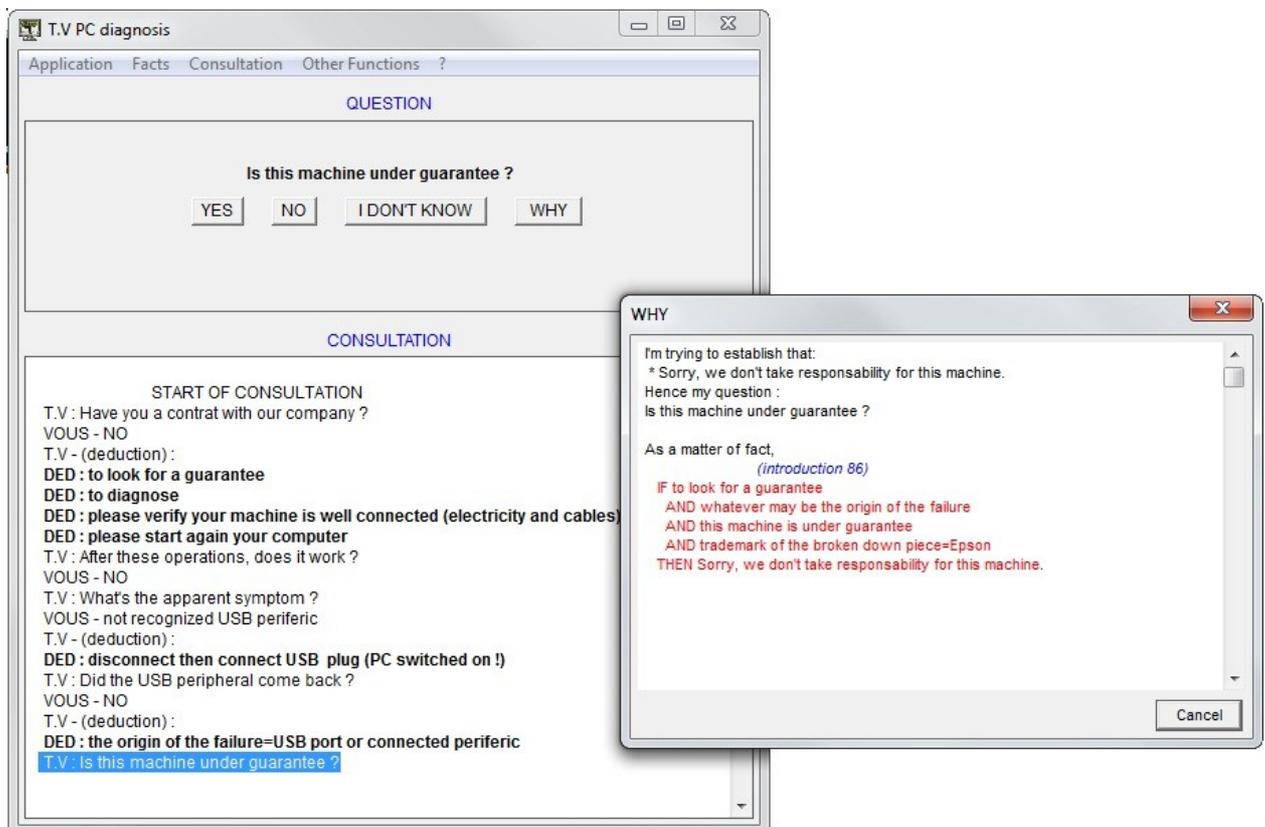
@ REGLE diagnostic 21
diagnostic 21
IF to diagnose
AND after these operations, it doesn't work
AND the apparent symptom="non working keyboard"
AND keyboard problem="inversion de caractères à l'affichage (par ex Q affiché à la place de Q)"
AND your OS="MS-DOS"
THEN please verify the DOS configuration files (autoexec.bat and config.sys)

@ REGLE diagnostic 22
diagnostic 22
IF to diagnose
```

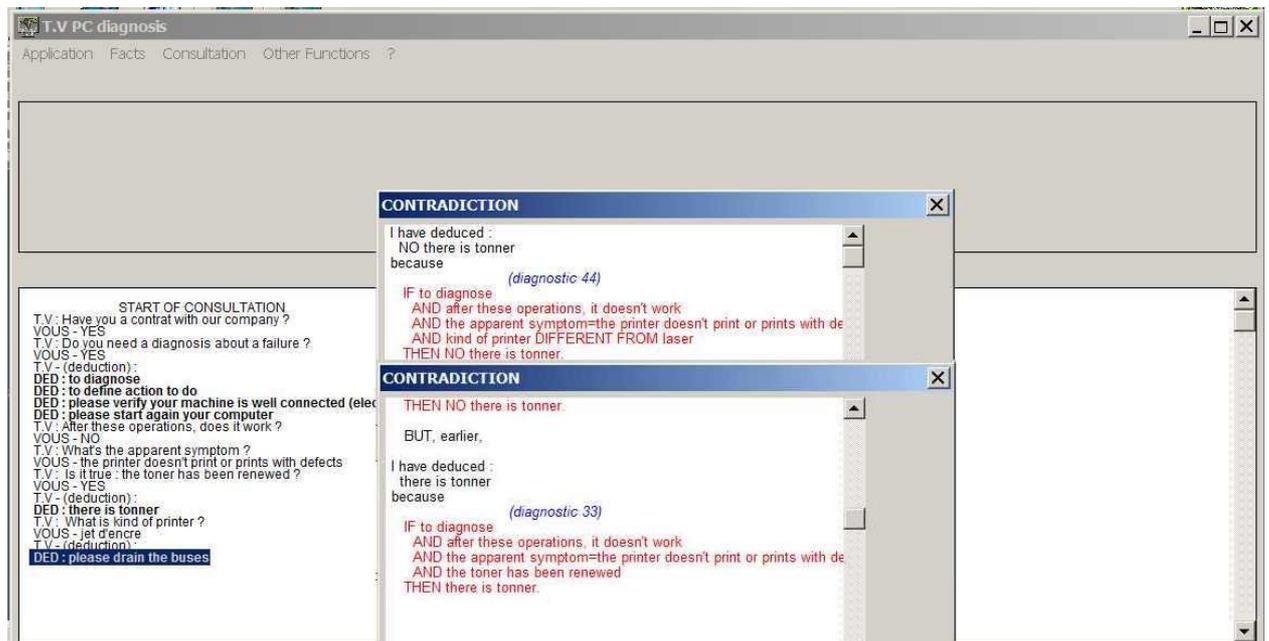
In fact, each branch is a rule, a rule is a branch. The knowledge base is automatically generated, rules are never written by a developer (the main source of errors). Subsequent maintenance is easily done by trees modifications, humans never touch the rules. La Maieutique checks consistency and completeness. It can also convert all the reasoning ways into one HTML procedural program (to cross firewalls or improve performance).



Verified rules Generated by La Maieutique are processed by a zeroth order engine, Moca, developed in 1988 by my computer engineers. Today it does not have any known bug. Moca produce a dialog guided by shorter way (the minimum of questions), explanations, contradiction detection, reasoning processing of "I don't know" answer (to automatically adjust the level of knowledge of the user and not tire expert users):



Here is a contradiction example:



This tool allows the development of types of applications impossible to achieve otherwise: logic simulation of systems, test of technical ideas, conversational and very interactive software, educational software showing operation of knowledge, explaining it and detecting contradictions of the student, constantly changing software.

This is how a machine produces fair reasoning and good conclusions using knowledge it doesn't understand... In sum, nothing out of the AI ordinary, as defined in the books but we never see. However, it seems I am the only one able to quickly develop real expert systems today.

Conversationnel: an essentially human skill

As you can see above, the corollary of this reasoning expert system is its ability to dialog with the user. We saw the first step with Mycin. I called this ability "conversational" in the 1990s. *Turing test becomes trivial*. My first expert system (1986), Joséphine, had more than 1000 rules and the customer-expert system dialogue could last more than 15 minutes (including customer reflection time). Passing the test only requires to imagine the possible dialogues, development will be almost immediate with la Maieutique.

Miao and Flow Logic : the power of reasoning in fault diagnosis

A privileged field for expert systems, everyone knows, is the fault diagnosis. My discovery in this domain is, thanks to reasoning, experts of the machine are not necessary to gather troubleshooting rules. *It's enough to read its plan* (and be a technician or engineer). Propagation of fault in any system is governed by laws: “*The six laws of Flow Logic*”. If you have the plan of a system, with these laws you can read how the default spreads and find its origin.

The six laws of Flow 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)*

Each component is described with a decision grid and not a decision tree. For example an electric pump:

		DOWNSTREAM		
FAULTS	UPSTREAM FAULTS	<i>Flow=none</i>	<i>Flow=low</i>	<i>Flow=too high</i>
	<i>Flow = unsolicited</i>			
	<i>Electric supply = 0 V</i>			X
	<i>Electric supply <150 V</i>		X	
	<i>Electric supply >250 V</i>			N
	<i>Electric supply = unsolicited</i>			N

X = the component may be the cause of the fault, if not it is upstream

N = the component can't be the cause of the fault, it is upstream

The power of this logic is astounding. Miao detects the faulty component with a very small number of questions and leads diagnosis ten times faster than a team of experts for the machine. Computer has a unique advantage over the man: its memory. It has

in mind the whole detailed plan of the machine, its deductions are safe, it doesn't hesitate and does not forget anything. With Flow Logic this little grid allows to automatically generate 24 rules in zeroth order logic:

- 1) IF flow = none AND electric supply \neq 0 V (ie DIFFERENT FROM 0V) THEN pump failure
- 2) IF flow = low speed AND electric supply \neq (<150 V) THEN pump failure
- 3) IF flow \neq none AND \neq low AND \neq too high THEN flow = normal
- 4) IF electric supply \neq 0 V AND \neq (<150 V) AND \neq (> 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 \neq low AND \neq normal AND \neq too high AND \neq unsolicited
- 9) IF flow = low THEN flow \neq none AND \neq normal AND \neq too high AND \neq unsolicited
- 10) IF flow = too high THEN flow \neq low AND \neq normal AND \neq none AND \neq unsolicited
- 11) IF flow = unsolicited THEN flow \neq low AND \neq normal AND \neq none AND \neq normal
- 12) IF flow = normal THEN flow \neq low AND \neq too high AND \neq none AND \neq unsolicited
- 13) IF electric supply = 0 V THEN electric supply \neq (<150 V) AND \neq (> 250 V) AND \neq unsolicited AND \neq normal
- 13) IF electric supply <150 V THEN electric supply \neq 0 V AND \neq (> 250 V) AND \neq unsolicited AND \neq normal
- 14) IF electric supply >250 V THEN electric supply \neq 0 V AND \neq (150 V) AND \neq unsolicited AND \neq normal
- 15) IF electric supply = unsolicited THEN electric supply \neq 0 V AND \neq (> 250 V) AND \neq (< 150 V) AND \neq normal
- 16) IF electric supply = normal THEN electric supply \neq 0 V AND \neq (> 250 V) AND \neq (< 150 V) AND \neq unsolicited
- 17) IF flow \neq none THEN electric supply \neq 0 V
- 18) IF flow \neq low THEN electric supply \neq (<150 V)
- 19) IF flow \neq too high THEN electric supply \neq (> 250 V)
- 20) IF flow \neq unsolicited THEN electric supply \neq unsolicited
- 21) IF flow \neq none AND \neq low AND electric supply = 220 V THEN flow = normal
- 22) IF flow \neq none AND \neq low AND electric supply > 220 V THEN flow = too high
- 23) IF flow \neq none AND \neq low AND electric supply = unsolicited THEN flow = unsolicited
- 24) IF electric supply \neq (>250 V) THEN flow \neq too high

This method can infer the state of flows without wasting time to test them. After each test we know the state of a flow therefore we know its other states are impossible. If a flow has 3 possible states and it is deduced two states are impossible, the third becomes true without testing. Therefore all the downstream components receive this flow in a well-known state which shortens the diagnostic.

Miao software has been sold and used in many companies and has been the subject of numerous articles in the press.

Tiara and voice T.Rex: the intelligent machine

Tiara ([here](#) an English presentation for European scientific data base) is a female chatterbot for PC. It displays in a cartoon bubble the text it says by voice. With Tiara, a PC instantly becomes capable of functioning by reasoning and vocally interacting with the user. It is equipped with voice recognition and speech synthesis (Scansoft) and can speak almost any language. It has voice commands logic for computer average user (not the commands imposed by Windows because it knows how to convert user needs into Windows instructions if necessary), that user can modify himself. In its current state it is a prototype that works perfectly but it uses 1999 voice interfaces that have bugs and must be changed. You can dialog vocally or silently by clicking in the bubble. Tiara has the basic knowledge of an IT specialist contained in an expert system of diagnosis of PC/Windows hard and soft failures and can help users interacting with them and prevent main bugs. Tiara also contains a *voice T.Rex* which draws itself trees by dialoguing with the user. It allows any user to develop complex (or not) applications from his personal expertise or that of his family. A child is able to use this tool. Expressions are not written - not recognized by the voice recognition - but replaced by [waves](#) with the user's voice. The expert system talks with user voice. In the current state, the largest part of the voice T.Rex programming interface is vocal. It is very easy to make it totally vocal. *We can program a robot on Mars, or modify its programs, only by dialoguing with it by phone from Earth.* Since the user has T.Rex, he can modify the failures diagnosis expert system and continually improves.

With Tiara, *PC can be used without a keyboard, without a mouse and even without a screen.* It can be stored in a closet, as a black box. To use it it suffices to install a microphone in each room and a screen in the main rooms. You can converse with him while doing something else and moving. You can even consult or program it by phone from the other side of the planet. It becomes able to continuously learn user knowledge of the entire planet, more and more complex and updated every day, and use in the interests of its users 24 hours a day. Each PC becomes different, a machine created by his master, his image.

You can see Youtube videos of Tiara here (in French, sorry):

1. [Tiara installation](#)
2. [Talk with Tiara](#)
3. [Tiara library: Bibtara](#)
4. [Self learning and programming tool Tiartrex](#)
5. [Voice Conversationnels examples written with Tiartrex](#)

Maieutica, Moca, Miao and T.Rex were programmed primarily in Prolog but might as well be written in Pascal or any other language because in fact Prolog offers no faculty for the development of a reasoning tool. Moca is also developed in Java by Jean-Bernard Wilhelm.

Intelligent machine - conscient machine

In a 2008 article about [reasoning AI](#) (in French, sorry) I show how Tiara is a brain simulating human faculties except invention faculty (inductive logic). Given that the computer/web memory capacity greatly exceeds that of man and its ability to learn is much more extensive and reliable, the produced intelligence exceeds automatically his own intelligence (except in creativity domain). Initially on short themes but quickly, with time, becoming a real general culture.

In a 2009 article about [artificial consciousness and robotics](#) (in French...) I show with examples how easy it is for an android robot to give the illusion of a human consciousness. Consciousness is not a human specificity; it is the fact of speaking about consciousness that is a human specificity. Just as intelligence, consciousness is not a human specificity. It is one of the basic mechanisms of the life and is found even in unicellular plants and plants.

Consciousness is the instrument panel of living creature. It contains all the environmental data perceived by senses on which intelligence reasons and decides. The computer has too few sensors and actuators to be able to offer evidence of consciousness. But a humanoid robot can have a large number of sensors and actuators. With reasoning AI it can easily simulate a human conscience.

If a robot gives all signs of consciousness, dialogues with relevance, is it really conscious?

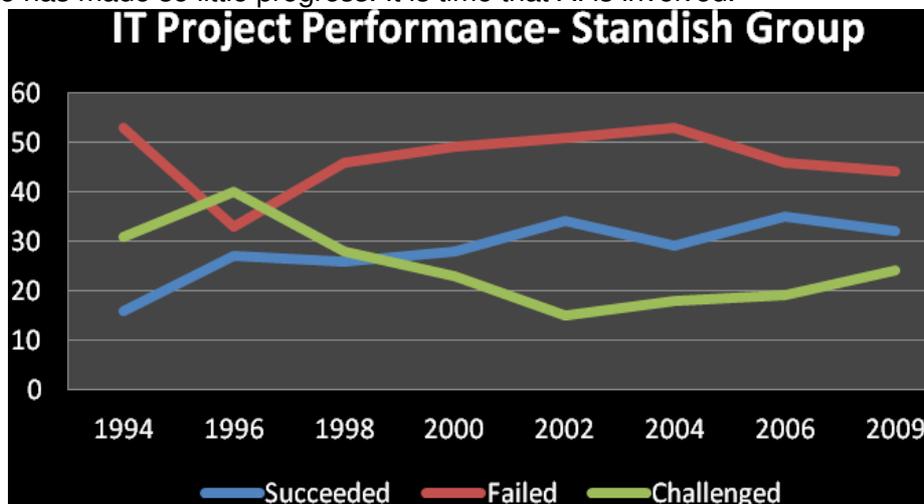
If a human gives all signs of consciousness, dialogues with relevance, is he really conscious?

Prospects

I had time to think about the computer of the future. My first remark is, as it is today, it contains Boolean algebra in its processor therefore all the logic and arithmetic instructions necessary to a reasoning engine like Moca. The computer is natively intelligent without resorting to programming. Therefore any application can be programmed without computer language.

My second point is that, when it comes to program an algorithm, whichever way you look at it, a complex language comprehensible only by IT specialists is completely useless. One basic instruction be enough. Its role would be to instantiate a list of text rules. So would a micro expert system. All the rest of the application will be written as text rules in everyday language in order to execute the desired behavior. The level above, developer (or expert) provides a list of rules that explain the batch reasoning (non-interactive = all entry data known from the beginning). Level above he adds to this batch engine new rules to explain how to conduct a relevant conversation (Conversationnel). Above, new rules describe how to conduct the interview of an expert and synthesizing rules (Maieutica). And level above there are business rules. If the expert wants only a batch application, such as payroll for example, the conversational level is not used.

IT developers program the same way since 1950! They follow the antique principles of the Turing machine inspired Jacquard looms. They describe the computer step by step what to do, an enormous task producing full of human errors instead of explaining what to do, trusting its reasoning ability. Even worse, they must first discover and understand each expert knowledge, another superhuman task. The result of this coercion is a stagnant productivity for 60 years! Today only a third of software meets customer demand, a third is thrown, last third must be modified. "Software Crisis" is a famous social phenomenon. No technology other than computer science has made so little progress. It is time that AI is involved.



Software crisis: Bankruptcy of programming methods

Examples of Conversationnels developed with Maieutica/T.Rex and Miao (by experts and a collaborator of Mr. de Lespinay, without computer specialist)

- [Joséphine](#) expert system in financial investments (Banque de Bretagne, 1986)
- Conversational user guide for Intelligence Service/Pandora (GSI-Tecsi, 1987)
- Expert system of PC hard and soft troubleshooting (Thomson Group, 1987)
- [Createst](#), advisory expert system to new entrepreneur with analysis of the candidate and writing a custom report (National Agency for entrepreneurship, 1988)
- Automatic monitoring of water treatment for ELF Donges oil refinery (1989)
- Aided design of electronic circuits (SMT Microelectronics)
- [Exportest](#) aid to the new exporter with analysis of state subsidies (Region of Pays de la Loire, 1990)
- [Aloes](#), orientation aid in higher education with an analysis of the student, proposal of professions and corresponding training (Nancy 2 University, 1993)
- Soudfe, aid in welding electron beam (French Navy)
- Procedural push emailing generated from commercial rule base (Arcane)
- Troubleshooting gas boiler [Saunier Duval](#) (1990)
- Troubleshooting inverters [Merlin Gérin](#) (1999)
- “Virtual Customs Office” by the web for French people wanting to import their goods into France (Ministère des Douanes)
- Intelligent vocal server giving advices about credit requirements to purchase a home ([Meilleurtaux.com](#))
- Emergency medical conversational diagnostic by phone ([Clic System](#), 1998-2000)
- Creation and test of Credit Scoring knowledge for corporate clients ([Sovac](#) Bank, General Electric subsidiary)
- 100 educational versions of Miao and Maieutica for French National Education (for secondary school and technical training)
- Contradiction tests in future legislative texts (Ministère des Affaires Sociales)
- Assistance in sale of laboratory equipment for beginner sellers ([Jouan SA](#), 2000)
- Knowledge transfer in the design of automotive parts ([Legris Autoline](#), 2000)
- Reasoning script for call center operators ([High Co](#), 2001)
- Payroll batch reasoning expert system (Jean-Bernard Wilhelm, 2011)
- Etc.

I wrote in 2011 most of Wikipedia's [Expert System](#) article that can be regarded as the continuation of my French [article in Science & Vie](#) 20 years before (1991).

My influence was great in France from 1987 until 2002 (in 2000 I got the FCPI State label of innovating company for the Tiara project). Until today hundreds of press articles talk about my R&D and my implementations in companies. Unfortunately, French "official" science is a fierce enemy and always refused to talk bustling everyday in order to force my silence by the most incredible sabotages. In 2002 they succeeded, forcing me to close my company, ruining me. *"When a true genius appears in the world, you may know him by this sign, that the dunces are all in confederation against him"*, Jonathan Swift (I never said I am modest).

Proposed Citation

"For high-impact contributions to the field of artificial intelligence through innovation and achievement in knowledge extraction, automated reasoning, conversational, self-programming, and making AI technology operational and accessible to the general public."

References (apart from [list of articles](#) in newspapers)

Michel Le Seac'h

Reference Name: Michel Le Seac'h
Address: 2 rue François-Joseph Talma
City: Nantes
State:
Zip or Postal Code: 44000
Country: France
Email: Leseach@aol.com
Date January 28, 2013
Phone : (33) 240 74 69 49 and (33) 660 73 66 67

Dear Sirs,

I worked with Jean-Philippe de Lespinay as a « maïeuticien » for about two years in 1986-1988. I had no technical background in IT or AI. After studying law and political science (I graduated from the Institut d'études politiques de Paris, usually called « Sciences Po »), I worked for a middle-sized French bank as an aide to the Company Secretary (7 years) then as a sales manager in their IT department (1 year).

De Lespinay's company, Arcane, recruited me to develop a banking expert system, *Joséphine*, using their "Maïeutique" methodology, which meant I had to interview two experts from a French regional bank, the VP of private investment and his research assistant, and shape their knowledge not as a regular IT analysis leading to code development but *directly as knowledge rules in plain French language*. Actually, while I had no technical background (or perhaps even *because* I had no such background), this job matched rather well my previous experience as one of my main assignments had been for 7 years to keep record of bank executive decisions. (A second match was that I had a working knowledge of PCs, which was not common at the time amongst non-technical people.)

The three-month development of *Joséphine* went well although the tool we used, Intelligence Service, was still not perfect at the time. While AI experts were devising grand theories saying the expression of expertise had to be complex, the 2 experts and I discovered simplicity was the rule if we were to develop an operational expert system, thanks to an intuitive decision-tree process. Still, we were able to get to a reasonable degree of granularity in the rules. The number of rules became important (around 1,000 rules) but we could limit the complexity of the system by using simple meta-rules (for example, IF a customer showed such and such conditions, THEN he would qualify as a "wealthy customer", and then the reasoning would go on about the way a "wealthy customer" should be treated). These meta-rules were treated as normal rules. Also, calculations were developed separately in Pascal then integrated with *Joséphine*. This expert system was the first one used in a French bank as a front commercial tool directly used with customers, most of them totally ignorant of IT. It got a lot of press coverage at the time.

After completing *Joséphine*, I worked on several other expert systems, including one about software servicing for an important French financial software company, which eventually recruited me to work in their development department. I left after about 1 year to become a free-lance writer (with this company as my first customer – I wrote their quarterly journal for several years). By the way, the first important piece I wrote was a full book, *Comment développer des systèmes experts* ("How to develop expert systems") (PSI, Paris 1999). Then, writing, translating and publishing assignments took me out of the specific field of AI.

However, I have been maintaining a friendly relationship with Jean-Philippe de Lespinay. I admire him much for the commitment, single-mindedness and personal energy he invests in his enterprise. Well, perhaps even too much commitment and personal energy sometimes, which means his intensity may irritate the “regular” IT scientists and the academic world in general. But after all, great accomplishments always need passion!

Sincerely,

Steven Morvan

Reference Name: Steven Morvan
Address: 1 Allée des Iles Gambier
City: Nantes
State:
Zip or Postal Code: 44000
Country: France
Email: smorvan@sigma.fr
Date: January 28, 2013

Dear Sirs,

I'm formally an IT engineer, acting as Innovation Manager in a large French software and IT company ([Sigma Informatique](#)). AI has always been a topic of interest for me. I even started a Science PhD in 1995, at [LINA](#) lab in Nantes, France. I decided to leave it : research works were only about updating old theories, rewriting old AI engines in modern or exotic languages... I couldn't expect any breakthrough. I met Mr De Lespinay by chance in 1998, and immediately felt that his approach was revolutionary. It is sensible, practical, and usable by anyone who can explain his way of thinking – that made quite a difference!

I eventually joined Sigma Informatique (a thirty years old company is focused in writing financial and retail software, that is, in principle, not the best context to experiment AI tools) in 1999 as a network engineer. I started to spread the word about T-Rex. We bought our first T-Rex licenses in 2001 and had several proofs of concept. It took me almost 10 years to convince some colleagues that a different, non-traditional, way to conceive software could exist. However I managed to deploy Mr de Lespinay's solution, in various ways such as:

- Help pre-sale with a *Conversationnel* tool to ensure that quotations were complete, by modeling the sales expert way of conceiving quotations.
- Increase hot-liners efficiency and capabilities, by guiding them in the questioning process. It allows them to treat incoming calls thoroughly and transmit well qualified calls to the experts. It really gives us a competitive advantage in client's relationship management. We acquired 6 licences by the end of 2012 and are currently deploying this solution.
- Model our clients' business process in an efficient and sensible way. The model is written in natural language, it is easily shared and understood by non-IT persons. Then, we use it as a reference to write our software with traditional means.

New ways of using the concept appear monthly, whenever we need to share more than data or knowledge.

From a company entrepreneurial point of view, Maieutique is priceless whenever we need to share know-how. Know-how that is, in fact, our real critical asset. ROI is easy to prove and quick to reach. But, as an early adopter, I can testify that it requires an efficient change management. It can require much time – nearly as much time as we save using Mr De Lespinay solution. Why? Because, even in 2013:

- IA is still believed to be something out of reach for most people, reserved to a few. We still have to convince people that AI is not science-fiction. Publicly recognizing and supporting Maieutique would greatly help spreading the concept of AI.

- Mr De Lespinay has guessed almost 30 years ago that user experience is central to the adoption of a technology, and, like other disruptive innovations, IA solutions must be driven by the usages rather than the technologies.
- Sharing knowledge is a great achievement of the Internet era through wikis, etc. Now, I think the next great step (not to say singularity) will be about sharing know-how, and make it available to everybody, especially people without IT skills. IA is the ideal solution for that. IA researchers should be the pioneers, by popularizing IA, accepting and promoting non-academic contributions such as Maieutique.

Sincerely,

Jean-Paul Baquiast

Reference Name: Jean-Paul Baquiast
Address:
City: Paris
State:
Zip or Postal Code: 75
Country: France
Email: jp.baquiast@wanadoo.fr
Date: January 23, 2013

Dear Sirs,

The non-commercial online journal Automates Intelligents.com lists for 13 years the main innovations in emerging science and artificial intelligence. Its editors had known Jean Philippe de Lespinay since a long time. This is a specialist IT and artificial intelligence who has made a large number of tools and interesting works in this area. He made AI a daily reality with many installations in companies. He proposed a decision trees method for the extraction and exploit of experts knowledge, following the principles of procedural memory. It is the Maïeutique...Jean-Paul Baquiast et Christophe Jacquemin, pour Automates Intelligents

Sincerely yours

Jean-Paul Baquiast

www.automates-intelligents.com
www.admiroutes.asso.fr
www.europesolidaire.eu/cont.php

Other testimony on the website of Jean-Paul Baquiast, [Dossier The reasoning Artificial Intelligence](#), December 21, 2008 (translation) :

The editors of Automates Intelligents know Jean Philippe Lespinay since at least 20 years. This is a computer specialist and artificial intelligence (AI), which has produced a large number of tools and interesting work this area. It is now estimated that the AI stalled due to lack of scope and ambition of the vision. New products could be made relatively easily according to him. This would still require the minimum conditions for their development are met, which is not the case in France today. He honors our magazine thinking it can promote widespread awareness of these opportunities thanks to its large diffusion and fueling a debate. He has kindly entrusted the publication of what we would call a founding document on the prospects of a new IA.

he honors our magazine thinking it can promote thanks to its large diffusion widespread awareness of these opportunities, for raising debate. He has kindly entrusted to us the publication of what we would call a founding document on the prospects of a new IA.

We believe it would be desirable to organize a debate on these issues. The document can elicit reactions, "for", "against" and "other." We will implement a conversational about it after the holidays.

Jean-Paul Baquiast et Christophe Jacquemin, for Automates Intelligents

Bertrand Tavernier

Reference Name: Bertrand Tavernier
Address: 8, rue des Charmes
City: La Chapelle sur Erdre
State:
Zip or Postal Code: 44 240
Country: France
Email: tavernierbb@orange.fr

Date: jan. 22nd 2013

Dear Sirs,

In 2000 I was acting as commercial manager France, in JOUAN SA company. Jouan used to design, manufacture and sell laboratory equipments, in direct and also via channel partners.

- In the end of 90', the company was facing issues to get customer service front office people permanently aware and trained on equipment our company offered.
- After benchmarking several solution, we finally worked with Mr De Lespinay as the system he was promoting (T.Rex) is fast and easy to implement, and even more, we got 100% of ownership on both licences, technical knowledge, training capabilities and data update. This would ensure us with total cost of ownership.
- T.Rex setting and management is designed to be held by non IT expert person : This is a clear added value as the project pilots then can be business manager, commercial leader or whoever in direct connexion with goals. No loss of time fighting with IT because of various priorities.
- And this was confirmed. After no more than 2 days of training for project drivers, it took us roughly 2 months to get our hot line team at higher level than project leader. And I would say that after 3 months time, customer service employees knew the product so well that they did not use T.Rex anymore for standard question, just using it as a confidential support for accessories, specific features and questions.

In conclusion, T.Rex is a perfect tools to get people concentrated on business, not taking care of way to get information.

I would qualify this as a management tool, which needs to be appropriated by users in order to ensure it is easily up dated by users themselves.

Sincerely,

Bertrand Tavernier

Jean-Paul Souris

Reference Name: Jean-Paul Souris
Address: 5, Rue de Genets
City: Adainville
State:
Zip or Postal Code: 78113
Country: France
Email: contact@jpsconsultants.com
Date: January 28, 2013

Dear Sirs,

I am a cybernetic engineer acting as a Consultant in Maintenance. I use a method of problem solving, Maxer, based on causal chains composed of OBJECTS / DEFECTS. I was also trained in Master Black Belt 6 Sigma by Mikel J. Method Harry, designer of this method at Motorola in 1996.

You guess that my job is focused on fault diagnosis and system reliability it highlights the fantastic resources of reasoning. In Method Maxer, there is a chain of cause and effect, but to date visualized by graphics software, successive trees you must read, a way fairly restrictive. During 40-year career in this community of causes research, I looked for applications of interactive diagnostic if possible based on expert faire une démonstration de systems. According to the expert system theory, these programs are supposed to think and talk, and therefore be very effective for fault diagnosis. Unfortunately, among those that I could evaluate which flourished during the 1990s (MAINTEX of Framentec, Polymaint Expert Polymont, Diagnostic Wizard Cleversys, Amidoc of Calladan, Diag Cody Cody, IA Concept, Sisyphus-Essential Genilog, etc.) only JP de Lespinay's expert systems produced a real reasoning. Moreover one of them produced true fault diagnosis reasoning as a good technician does: Miao.

I met M. de Lespinay at a conference in 1992 when he came to me and saying "Hello, you are the Pope of maintenance and I'm the Pope of AI. We could perhaps work together? " He made a demonstration of Miao and, like all those who have approached this software I was impressed by its simplicity, power and clarity of deductive reasoning. This application from another professional environment than Maintenance looked rather like an UFO. Miao can even provide the fault diagnostics before the manufacture of the machine since it works "on plan." I immediately felt that his approach was innovative compared to my previous experiences. It was more accessible, practical and usable by anyone, while other applications were rather used by engineers who lacked natural intelligence ... In my opinion Miao would be difficult to sell to manufacturers, who are more used to record facts to look for causes.

Maxer method, as indeed Miao, have not found a wide audience when they appeared on the market because too ahead of their time. The only common point between two products is that diagnosis works mainly on reasoning. New ways of using the method Maxer (originally for maintenance technicians and now used by operators) appear monthly, whenever we need to share more data or knowledge. I think that the application of MIAO for the design of Maxer détaillogramme (failures diagram) can be a great plus. It allows the design of a feedback database smarter thanks to the invaluable concepts of Maïeutique in all cases where we must share know-how (which is in fact our true critical asset).

JP de Lespinay guessed almost 30 years ago that the membership of the user is the necessary starting point for the adoption of technology, and, like other disruptive innovations, AI solutions must be guided

by usage rather than by technology. Lespinay applications are likely to anticipate all potential problems at the stage of designing systems thanks to dynamic reasoning what AMDEC method does not allow. It is based on grids to complete and therefore particularly static. That gives us hope for future years.

Sincerely

Jean-Paul Souris

Dominique Bayle

Reference Name: Dominique Bayle
Address: 3 rue de la Barralerie
City: Montpellier
State:
Zip or Postal Code: 34000
Country: France
Email: bayledominique@free.fr
Date: January 28, 2013

Dear Sirs,

I founded a company in 1993, "HOO Technologie", specialized in developing software solutions on interconnections phone/computer (CTI). My team consisted mainly of computer engineers specialized in telephony, networks, databases. Scripting tools were integrated into our software, allowing operators to have a permanent "guide". As part of my commercial offer we proposed solutions "turnkey" including, in addition to our software (Management of incoming and outgoing calls, including the synthesis and speech recognition, combined a unified messaging) the all necessary equipment for call centers or contact centers (offices, computers, servers, wiring, computer connections, etc.).

JP de Lespinay and I had our companies in the same city, Nantes, and he was also targeting the call centers market. In 2000 he had developed a version of his Maieutica software called "T.Rex Call Center" for the development of scripts. These scripts can be text but also with voice recognition and voice synthesis. They allow to develop conversational voice server, a revolution. He contacted me and showed me T.Rex Call Center. I discovered for the first time in my life a program based on reasoning, able to communicate with users as an expert, and with a programming of the scripts completely intuitive. Seduced by this tool, I agreed to incorporate in my product catalog and I proposed it to my clients.

A few months later JP Lespinay brought me a client: the "B.active" call center in Paris, a subsidiary of High Co Group. The client wanted urgently to create a call center for 15 operators using T.Rex Call Center. We had the contract only if the call center was installed in one day and the script was ready to use by operators the next morning! We accepted the challenge and signed the contract. JP Lespinay developed the telemarketing script, an one hundred rules expert system. Operators should explore former clients in the field of tourism, arguing from the data stored about them in a database. In one day we managed this feat. The next day the call center and 5 telemarketing operators prospected clients assisted by T.Rex script. We were very proud of this unique feat.

In 2003, I sold my company to get into another project but Jean-Philippe and I have always remained friends. It is a nice man and full of humor and I always enjoy chatting with him via email or phone.

I think he discovered the only true demonstration of what must be artificial intelligence: a computer smart enough to program itself, that is to say to learn and use our knowledge to our service. And as far as I know, he is the only one in the world to reached this result. We can say, given its original formation which is that of a seller, it is a "genius". I hope he will win your prize because it deserves to receive advertising and everyone knows that with him a new era of computing opens.

Best regards

Dominique Bayle

Jean-Bernard Wilhelm

Reference Name: Jean-Bernard Wilhelm.
Address: 38, rue du Voulien
City: La Trinité sur Mer
State: Bretagne
Zip or Postal Code: 56470
Country: France
Email: jbw44000@gmail.com
Date: February 13, 2013

Presentation

My name is Jean-Bernard Wilhelm. After an engineering training I started my career in software development and project responsibility in several industrial and service companies (Cap Sogeti). Then I became a sales engineer at IBM where I worked with industrial and services accounts in the western region of France. Since 1986, I work as an independent consultant in information systems, with an increasing organizational part.

First contacts with Maieutica

In 1995, I met Jean-Philippe Lespinay. He introduced me Maieutica. The concept and the tool seemed to me very attractive. The simplicity of recording expertises and especially testing them surprised me a lot. At the time the use of the tool was essentially in "stand alone".

My first impression, which has not changed since was that this technology could be usefully integrated into the tools I knew, essentially for operating or business rules (business objects, application kinematics , complicated HMI ...) for professional management applications.

Indeed, object-oriented programming coupled with event programming had the main failing to explode logic towards many methods, making applications difficult to maintain. At that time, I used to describe a program as an assembly of several types of components: HMI components, components for two-way access to data, business components, a component of application kinematics and technical objects. Unfortunately, the mere object-oriented technics did not allow to realize easily this kind of components without dispersing the logic in all objects.

So I saw with Maieutica and its rules engine the opportunity to refocus the intelligence of program into external expertises outside the program, thereby combining the best of both worlds (object and expert system).

Cooperation with Tree Logic

First in 1995, at a time when I had the availability, I worked with Jean-Philippe de Lespinay on client presentations in IT services, with the goal of integrating this technology into client computer systems. This period lasted about six months.

On two occasions I considered buying Maieutica technology with a non-exclusive contract for an amount of € 500 thousand. However while functional teams of companies were very interested the projects were blocked by their IT . . .

Later in 2008, we worked together again to port in Java the rules engine of Tree Logic. I piloted the project, prepared functional specifications in collaboration with Jean-Philippe de Lespinay, and performed the tests. Today, the engine is fully functional in both batch and interactive.

Technology interests

For me, the main interests of the technology are :

- Business analysis will be made by a business expert, tested and integrated into an application without the need for IT specialist to understand the business part. The computer will consider this procedure as a black box, which it must know the inputs and outputs to ensure the connection to the computer system.
- Application maintenance for business parts supported by Maieutica is also simplified since maintenance can usually be done by the business expert.
- The rules design from decision trees. These decision trees are only used to write a coherent rules-based and disappear at runtime. This point saves a lot of time compared to writing rule bases from unit rules (operating mode of current commercial engines), particularly for rule bases of substantial size. Furthermore, Maieutica has tools to detect contradictions (not in a formal way but at the time of execution), which helps ensure the consistency of the rule base, if the tests are exhaustive enough.
- The engine response times is acceptable. For example, with a standard laptop PC in 2009 and a base of 200 rules, response times were of the order of 18 ms, and can still be improved in the case of batch expertises.

Perimeter of the technology

It has always seemed to me that the position of Jean-Philippe de Lespinay of eventually replacing programming languages by a tool like Maieutica/T.Rex is untenable, at least in the current state of technology.

This tool is great for treating procedures, especially complicated procedures but he does not know the notions of data structures because it works with flat data. Especially in management information it is necessary to provide integration with databases.

Problem of acceptance by IT professionals

The big problem we have encountered is the difficulty to make the IT specialists understand the interest of this technology. It seems that the main reason is the fear of being deprived of part of their power. In addition, resistance to change acts with strength in IT as in any population. Finally, it is not impossible that IT see in this technology an aspect "magic" that causes considerable distress to them. I see no other explanation for the emotional reactions that we have seen.

Example of use

I personally developed a standard payroll program with Maieutica. French Payroll is particularly complicated. It is perhaps one of the most complicated in the world with that of Belgium.

I managed to write a payroll engine operational after two days for the part payroll rules. This engine is capable of handling all the standard part of a payroll, ie the legal part that is common to all types of business. The rule base contains 245 rules and processes 252 different types of facts. I think that in classical algorithms it would have taken between five and ten times longer to achieve. In addition, the maintenance of an expert system written with Maieutica is much simpler than a conventional program, to the extent that maintenance is done directly on the trees and, most importantly, it can be done by a business expert. The gain in maintenance is even greater than the gain in coding.

Conclusion

For me Maieutica artificial intelligence technology remains a promising technology that could greatly serve both project managers and internal clients, significantly reducing misunderstandings between these two functions, or even by avoiding them because developers would not have to become themselves business experts.