On the Physical Formal and Semantic Frontiers Between Human Knowing and Machine Knowing

José Mira Mira

Dpto. de Inteligencia Artificial,
ETSI. Informática. UNED, Madrid, Spain
jmira@dia.uned.es

Abstract. The purpose of this paper is to reflect on the nature of human knowledge and that of the knowledge that finally can dwell in an electronic computer. Three frontiers can be distinguished between these two constitutively different types of knowing: (1) The nature of current physical machines (silicon semiconductor crystal) and its organizational restrictions in relation with the biological tissue, which is autonomous, dynamic, tolerant to failures, self-organizative, and adaptive. (2) The semantics of the available algorithms and programming languages in relation with the evolutionary and reactive (behavior-based) biological programming strategies. (3) The nature of current formal tools in relation with natural language.

1 Problem Statement

A great part of the AI community attempts to interpret and use concepts of the computational paradigm when applied to natural systems as equivalent to the corresponding concepts when applied to artificial systems. These scientists assume that natural language can be reduced to formal language, that biological programming strategies can be reduced to conventional programming languages and that the nervous tissue and a body of meat can be reduced to a CPU of Silicon crystal and an electro-mechanical robot. Unfortunately, fifty years after the christening of AI at the 1956 summer conference at Dartmouth Colleague, this is not the case. There are relevant, constitutive, differences between Computation in Natural Systems (the human way of knowing) and Computation in Artificial Systems (the machine way of knowing). To contribute to the establishment of a clear distinction between these two different ways of knowing we introduce in section two the methodological building of knowledge [11,9,10], which will enable us to distinguish between three levels and two domains of description of the knowledge involved in a calculus. In sections three, four and five, making reverse engineering, we consider the knowledge that each one of these levels can accommodate, from the physical level to the symbol level and, finally, to the knowledge level. Then we conclude, mentioning the topics in which, in our opinion, we should concentrate our efforts to move the frontiers between human intelligence and machine intelligence.

R. Moreno Díaz et al. (Eds.): EUROCAST 2005, LNCS 3643, pp. 1–8, 2005.

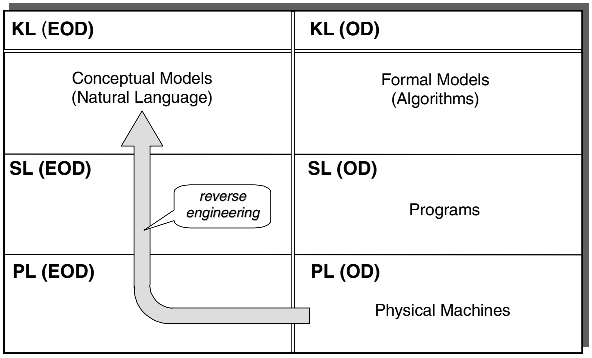

Fig. 1. Reverse engineering in the building of knowledge

2 Direct and Reverse Engineering Inside the Building of Knowledge

The three levels (storey) of the building for knowledge are the physical level where lives the machine hardware, the symbol level where the programs live and the third level, introduced by Allen Newell [12] and David Marr [4], located over the symbol level and named by Newell the knowledge level and by Marr the level of the theory of calculus. Additionally the external observer of a calculus (either in natural or in artificial systems) always can distinguish between two domains of description in each level [6,15,8]: the level's own domain, OD, and the domain of the external observer (EOD), as shown in figure 1.

In direct engineering we start with a conceptual model at the KL and in the EOD of the method used by humans to solve a problem. The architecture of this model depends on the AI paradigm used in our approach (symbolic, connectionist, situated or hybrid). Then we use a table of correspondences to move from this conceptual model to the abstract entities and relations of a formal model situated in the OD of the KL (the 3rd right apartment). Next we program the formal model and a compiler ends the work by producing the final machine language version of the program.

In this formalization process we have left out (in the third left apartments) a great part of the human knowledge used in the conceptual model. This non-computable knowledge establishes the first semantic and formal frontier between human knowing and machine knowing. Only the formal model underlying natural language words enters the computer. The rest of the knowledge always remains in the natural language of the external observer, in the EOD. Obviously the external observer injects this non-computable knowledge when he interprets the results of the calculus (left hand bottom-up pathway in figure 1).

If we now attempt to recover the conceptual model from which this calculus has emerged we have to make reverse engineering, starting on the first right apartment where the program is running in a physical machine, going through the symbol level and ending at the EOD of the KL. In this reverse pathway we will reflect on the type of knowledge that can accommodate each level, both in electronic computers and in humans. Then we propose a comparison frame to enlighten the differences (frontiers) between human knowing and machine knowing. Two considerations are common to the three levels, both in natural and in artificial systems:

1. Each level (KL, SL, PL) is a closed organization and its knowing capacity is constitutively determined by the set of entities and relations that characterize this level as a distinguishable class. That is to say, by its language [5,7].
2. The interpretation (translation) of this language in each level is structurally determined by the architecture and the language of the level below. That is to say, the available programming languages determinate the interpretation of the natural language models and the available machine language determinate the interpretation of the programs. Finally, the available materials determinate the limits of machine languages.

3 The Knowing and the Physical Level (PL)

The constituent entities of a computer hardware (logic circuits) only enable us to establish binary distinctions (0, 1) on the true or falseness of a set of logical expressions. If we consider digital delays the PL can also accommodate all the knowledge related to abstract internal states and state transitions of a finite state automaton (FSA). Finally the superimposed organization of the computer architecture provides computation with the electronic mechanisms necessary to accommodate the knowledge that can be described using the machine language. In conclusion, the PL of machines can only accommodate (OD) the formal knowledge associated with logical expressions, FSAs and machine languages. The semantic tables of these abstract entities always remain at the EOD of the PL, outside the machine.

The capacity to accommodate knowledge at the PL in biological systems also is a consequence of the constitutive entities of the nervous system (ionic channels, synaptic circuits, neurons, neural assemblies, ...) and of the neural mechanisms that evolution has superimposed at the architecture level (oscillatory and regulatory feedback loops, lateral inhibition circuits, reflex arches, adaptive connectivity and routing networks, distributed central-patterns generators, ...) and the emerging functionalities (learning, self-organization, reconfiguration after physical damage, ...) characteristic of an always unfinished architecture.

It seems now clear to us where is the first frontier between human knowing and machines knowing: In the constitutively different nature of its physical elements, mechanisms and architectures (figure 2), and, consequently, in the differences between the emergent semantics of neural networks and the limited semantics of the formal descriptions of combinational logic and FSA.

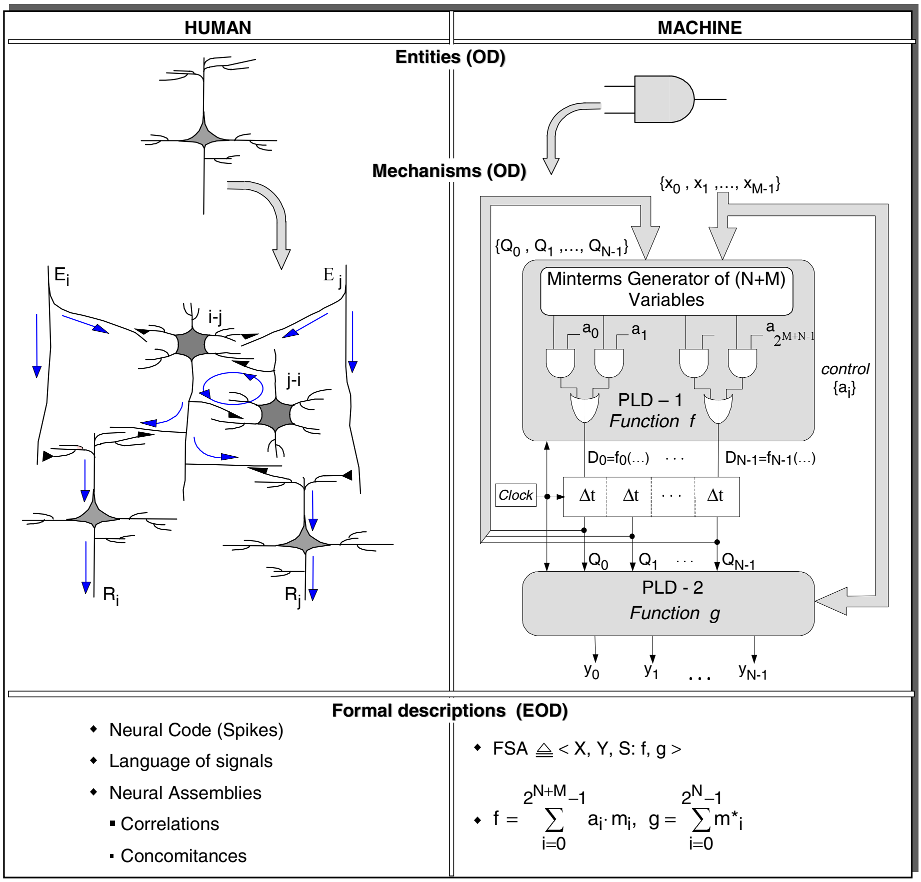

Fig. 2. The knowledge that can accommodate the physical level of human and machines depends on the entities (OD), mechanisms (OD) and formal descriptions (EOD) constitutive of this level

4 The Knowing at the Symbol Level (SL)

Let us assume that we know all the knowledge than can accommodate the circuits of the PL in humans and machines. We still do not know what the brain and the computer are calculating, until the description of the SL and the KL are completed. The knowing capacity of the SL is determined again by the set of entities and relations that characterize this level as a class (by its language). That is to say, the second part of the human knowledge involved in a calculus that finally can dwell in an electronic computer, is determined by the representational and inferential facilities and limits provided by current programming languages. The "Physical Symbol System Hypothesis" proposal of Newell and Simon [13] is representative of these limits (frozen symbols of arbitrary semantics, descriptive, abstract and externally programmable).

If we accept that evolution has accumulated enough mechanisms in the nervous system architecture as to accommodate what an external observer could describe as neurophysiological symbols and programs, then we can compare again the human knowing versus the machine knowing at the symbol level (figure 3). These symbols are connectionist and grounded in the neural mechanisms that generate and recognize them. Also, these symbols and the corresponding programs, are not programmable in the conventional sense, but via dynamic adjustment of a huge number of synaptic contacts. As a consequence, symbols in natural systems are emergent, situated, grounded and adaptive. Per contra, in artificial systems symbols are arbitrary, representational, static and externally programmable. These constitutively different characteristics establish the second semantic and formal frontier between human and machines knowing.

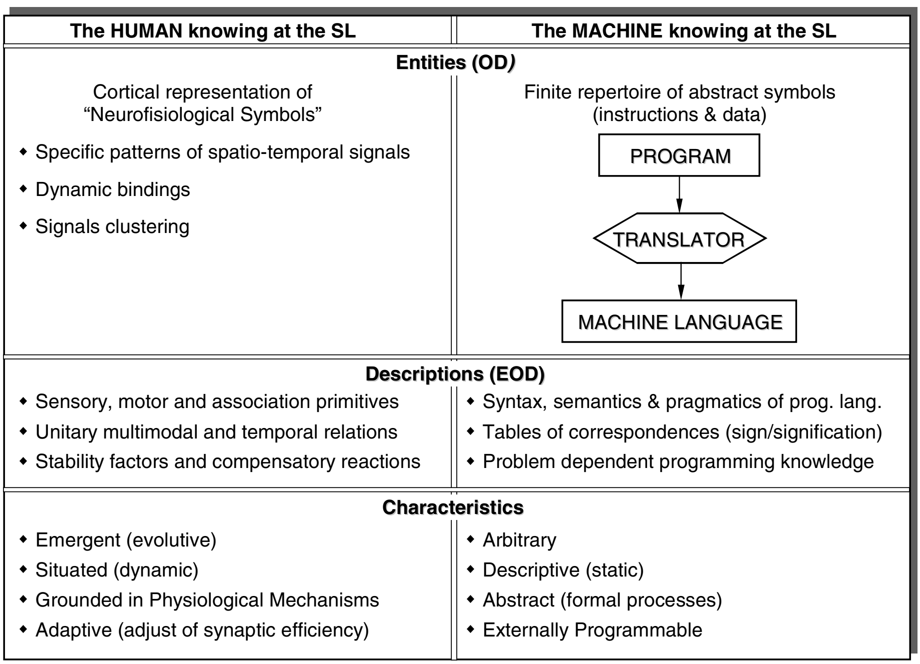

Fig. 3. The knowledge that can accommodate the SL in humans and machines

Some bio-inspired programming strategies (genetic algorithms, evolutionary and genetic programming) seems promising but are again for away from biology. Other approaches, such as computational ethology, collective emergent calculus (ant colonies, bees, ...) and the reactive (situated) approach to programming [3,2] are also trying to move the frontier between humans and machines at the SL.

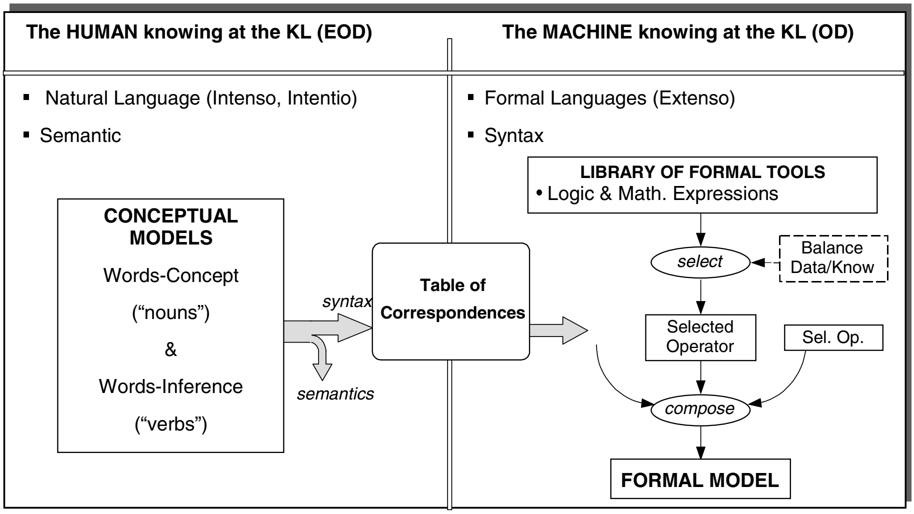

Fig. 4. Semantic frontier between natural and formal languages at the knowledge level. (Adapted from [10]).

5 The Knowing at the Knowledge Level (KL)

Let us finally climb upstairs from the SL (second floor) to the KL, the third floor of the building in which we have distributed the knowledge involved in a calculus. What are the differences between the knowledge that can accommodate at this level humans and machines?. Or, what is equivalent, what are the constitutive differences between the entities and relations of cognition and those of the conceptual and formal models currently used in AI and Knowledge Engineering? [14]. We do not know in deep the architecture of cognition but it is usually accepted that its constitutive entities are what we call perceptions, goals, purposes, intentions, ideas, plans, motivations, emotions, attitudes, actions, and so on. A major part of what we know about cognition has emerged through the observer's natural language. Consequently, we can consider that the accessible part of the architecture of cognition coincides with the architecture of natural language. Then the human knowing at the KL is the knowledge that can accommodate the natural language of the external observer. And here is the main frontier between humans and machines at the KL: Computing machines can only understand formal languages (the formal components underlying natural language models). In figure 4 we show a summary of the distinctive characteristics between these two types of languages as well as the current procedure in AI to reduce natural language to formal descriptions based on logic and mathematics. A table of correspondences between words and abstract entities is always necessary to store the meanings that are non-computable. These meanings are recovered when interpreting the results of the calculus.

6 Conclusions

The purpose of AI is to make human knowledge computable using conceptual models, formal tools, programming languages and electronic machines. Unfortunately, after fifty years and different approaches (cybernetics, heuristics, symbolic or representational stage, connectionistic again, situated and hybrid) this dream has not been yet fulfilled in a satisfactory manner. Two of the possible causes of this failure could be (1) the excess of initial optimism in assuming that cognition can be reduced to computation and (2) the lack of distinction between the constituent elements of humans and machines.

In this paper we have used the methodological building of levels and domains of description of a calculus to establish the constitutive differences at three levels (materials, symbols and languages) and we hope that the acceptance of these differences could help to delimit the real scope of AI and to focus where the real problems are: (i) Development of new materials and new computing architectures. (ii) Development of more powerful programming languages and algorithms and (iii) Development of new modeling tools and formal languages semantically closer to natural language, and still compilable.

Let us assume that these three objectives has been fulfilled. Would we say that human knowing has been then reduced to machine knowing?. Clearly, not. If the brain reasons as a logical machine the signals and symbols that it employs in its reasoning, must constitute a formal language [1], but it is not clear to us neither that cognition could be represented by using only formal languages, nor that the computational paradigm could be the only way to tackle living systems. What about cognition without computation?. Let us see what happens in the next fifty year of Neuroscience and AI.

References

1. J. Bronowski. The logic of the mind. *American Scientist*, 54, 1:1–14, 1966.
2. R.A. Brooks. Intelligence without reason. A.I. Memo 1293, MIT, 1991.
3. W.J. Clancey. *Situated cognition. On human knowledge and computer representation.* Univ. Press, Cambridge, 1997.
4. D. Marr. *Vision.* Freeman, New York, 1982.
5. H. Maturana. Ontology of observing. the biological foundations of self consciousness and the physical domain existence. http://www.inteco.cl/biology/ontology/, 2002.
6. H.R. Maturana. The organization of the living: A theory of the living organization. *Int. J. Man-Machine Studies*, 7:313–332, 1975.
7. J. Mira. Reverse neurophysiology: The embodiments of mind revisited. In R. Moreno-Díaz and J. Mira-Mira, editors, *Brain Processes, Theories and Models*, pages 37–49. The MIT Press, Massachusetts, 1995.
8. J. Mira and A.E. Delgado. Some comments on the antropocentric viewpoint in the neurocybernetic methodology. In *Proc of the Seventh International Congress of Cybernetics and Systems*, pages 891–95, 1987.

9. J. Mira and A.E. Delgado. Where is knowledge in robotics? some methodological issues on symbolic and connectionist perspectives of AI. In Ch. Zhou, D. Maravall, and Da Rua, editors, *Autonomous robotic systems*, page 334. Physical-Verlag. Springer, Berlin, 2003.
10. J. Mira and A.E. Delgado. From modeling with words to computing with numbers. pages 7–13, Budapest, 2004. Proceedings ISDA 2004.
11. J. Mira, A.E. Delgado, J.G. Boticario, and F.J. Díez. *Aspectos básicos de la inteligencia artificial.* Sanz y Torres, SL, Madrid, 1995.
12. A. Newell. The knowledge level. *AI Magazine*, 120, 1981.
13. A. Newell and H.A. Simon. Computer science as empirical inquiry: Symbols and search. *Communications of ACM*, 19:113–126, 1976.
14. G. Schreiber, H. Akkermans, and R. de Anjo Anjewierden. *Engineering and Managing Knowledge: The CommonKADS Methodology.* The MIT Press, Cambridge, Mass, 1999.
15. F.J. Varela. *Principles of Biological Autonomy.* The North Holland Series in General Systems Research, New York, 1979.