On Efficiency of Learning: A Framework and Justification

Jindřich Bůcha[1]

Abstract. A conceptual framework, whose goal is the improvement of efficiency of machine learning, is presented. The framework is designed in a broader context of problem solver (PS). The design is solved as an integration of all basic cognitive functions and as a software-engineering problem. Many (one hundred) requirements imposed on *PS* are considered. The most important of them are the object-oriented nature of the PS environment, reflexivity of PS, and central role of tool and shifted border.

1 INTRODUCTION

This is a position paper, presenting a framework for machine learning (ML), reacting on some reviewers' reamrks. The framework is very briefly described and justified. More details can be found in [1], [2], and [3].

My extensive evaluation of current ML reveals that the outstandingly dominant part of ML is devoted to simple learning. I consider this harmful and the framework is a trail to remedy this situation. I would roughly characterize **simple learning** as a one-shot creation of knowledge, describing one function of a small part of an environment, and applied in a manually selected part of environment. On the other hand, I would characterize **complex learning** as a cumulative creation of knowledge with many parts, which describes many related functions of whole environment. It is structured, preferably in object-oriented way, applied in different parts of different cognitive functions, applied in the whole real world. These are two extremes. The state-of-the-art is, of course, somewhere in-between, much closer to simple learning. It contrasts with some other AI areas, like knowledge engineering, where we do use complex knowledge structures [4], [5].

The application of this framework to **learning ontologies** is highly relevant: 1) Both in ontologies and in the framework, there is a common stress put on knowledge structure, reuse, object orientation. 2) Learning ontologies can be approached both top-down and bottom-up. If there are no worked out firm foundations then they should be done first. This is also the aim of the framework. 3) Both learning and ontology areas can be integrated. For example, learning could accept complex knowledge structures; ontologies could accept an approach to approximate knowledge. This would modify e.g. some design "principles" that "have been proved useful in the development of ontologies", like clarity, completeness, and coherence [4], [6]. On the other hand, this may explain e.g. one often-mentioned problem, that "concrete elements are in many cases practically more usable then abstract ones" [4], [7], [8].

My approach to learning can be characterized concisely as **"learn from learning"**. This means that it is suitable to use knowledge, which we have gathered during the exploration of learning (phenomenon), and apply it as meta-knowledge in the design of improved learning (algorithm). The body of this knowledge is vast. What parts should be used? My approach is to use **all meta-knowledge that can be integrated** into an efficient learning system. It is also the solution of the efficiency of learning: Each piece of this meta-knowledge should support somehow this efficiency. To design and build a corresponding system is a **formidable task**. However, realize four things: 1) There is a great redundancy in AI knowledge. 2) There is a difference between the building of a learning system and fully developed problem solving system. I am concentrating on **problem solving (PS)**, which has not yet gathered a massive amount of knowledge, but which can gather it. I am using the frame of PS to stress the importance of the context of learning. 3) There are approaches to the design of complex systems; I am using the software-engineering approach and the object-oriented methodology. 4) There is no evidence that this task is unsolvable.

The main **objection** against the proposed framework is why should we deal with something what is so unjustified, unproven, unimplemented? Haven't we done it already many times without any result? Is not here a rule that every proposal should be at least partially verified by some prototype? I understand this but do not agree. Why? 1) Critical claims are usually very vague. 2) Even negative experience is useful, it exists, and I'm trying to use it. 3) Prototype verification has its value; however, it should not be overrated. 4) Not only a detailed analysis, the synthesis can bring new knowledge too. To prototype a synthesized system is harder and needs more careful design. 5) It is in contradiction with the experience of software engineering: design should be verified step-by-step. 6) AI should realize the shortage of synthesis, its reasons and consequences, and should try to solve it.

In the design, I primarily use the viewpoint of efficiency of learning [9]. There are many other possibilities, e.g. the viewpoint of design of problem solver, integration of cognitive functions, design of agents, essences of intelligence. These viewpoints will manifest in a requirement analysis. This is usually the first design step of a software-engineering approach. Doing this, I have analyzed many projects, approaches and surveys, and I have extracted and classified key requirements either implemented or gained as conclusions from project experiences. I have gathered more then

[1] Institute of Information Theory and Automation, Pod vodárenskou věží 4, 182 08 Prague 8, Czech Republic, email: bucha@utia.cas.cz

one hundred **requirements**. Majority can be found in [3]. For some projects and approaches, I estimated the following fulfillment of these requirements: Minsky [10] 11%, CYC [10] 7%, PRODIGY [12] 18%, Soar [13] 23%, Brooks methodology [14] 33%, all these together 65%; my framework covers 100%. This is one reason why I consider the framework original.

2 FRAMEWORK

First, I am presenting basic assumptions about the learning **context**. They form a skeleton for the requirements:

- ***Environment*** is a *network* of many heterogeneous, dynamic and even uncertain *objects*.
- ***PS*** is an *object* of *environment*. *PS* has a pre-specified *goal*. If *PS* is not in *goal state*, it has *problem*, and that should be solved by *PS*. *PS* should control *environment* to reach *goal*. *PS* control is accomplished by means of its ***cognitive functions*** (*implementation*, *identification*, *reasoning*, *learning*, *knowledge base (KB)*, *self-control*, and *initialization*). The *cognitive functions* have also their (*sub*) *goals*.
- ***Knowledge*** is an approximate description (of the behavior) of *object*; *knowledge* usage increases the average probability of success of reaching *PS goal*.
- *Environment* is relatively very stable. During an interval of *PS* work, very little part of *environment* is changing.
- *PS* (*learning*) should cope with its complexity.

The framework itself is outlined in two figures. To fulfill the requirements, *PS* should be a knowledge-based system and *KB* should have the structure outlined in Figure 1. (For graphical notation see [15]).

Learning is not a stand-alone function. It cannot work without cooperation with other *PS cognitive functions*. *PS* should have the structure outlined in Figure 2. Notice that it is still a partial view, e.g. relations to important *self-control* and *initialization* functions are not rendered here. Notice also, there is only one **common concept**, *description*, for description of various *PS entities*, i.e. *classes* of *environment objects*, specific *objects* like *model*, *plan*, *value*, *variable*, *state*, *goal*, *meta-model*, *meta-plan*, etc.

3 JUSTIFICATION

Here I show two examples of requirements, tool and shifted border,

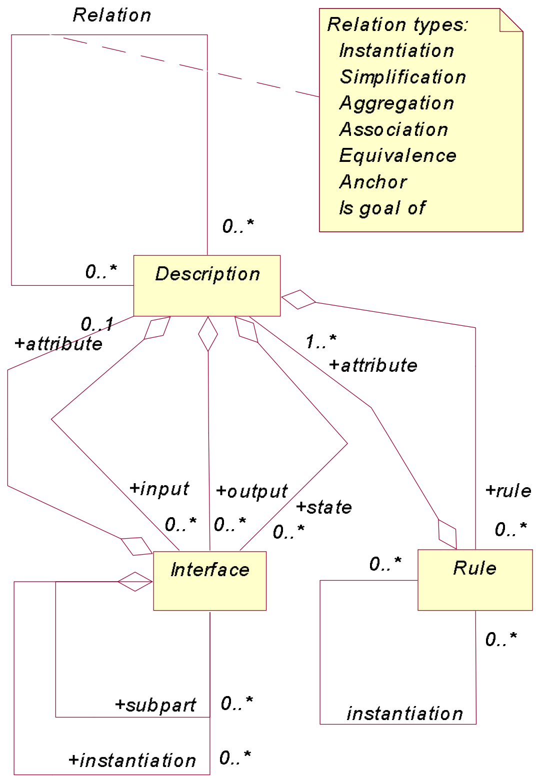

Figure 1. Model of knowledge base

and homogeneity, and briefly explain how they are implemented in the framework. More can be found in [1] where e.g. justifications cover differences between my approach and that of Brooks [14].

In a simple *environment*, to *identify* (the only one and simple) *object* and its *state* need not be a problem. To *learn description* of such *object*, *plan* a solution in this *environment* described as such *object*, and *implement plan* need not be a problem. *Self-control* would not be necessary and *KB* would be trivial. To implement it, the existing cognitive functions or ontologies from AI can be used [9], [4], and [16]. However, the real *environment* is not simple. It is necessary to extend the simple *cognitive functions* to cover the work with object-oriented *environment*, both deterministic and stochastic *objects*, with both static and dynamic *objects* etc. Some aspects of object-oriented nature of *environment* are solved by existing techniques, e.g. *instantiation*, *simplification*, and *anchor relations* in *learning*. To solve other aspects, e.g. *association relations*, the concept of *tool* is used: Let us consider *PS*, its *environment*, some **tool**, as an *object* of *environment*, and *rest of environment*. Let *PS* have *inputs* resp. *outputs* to perceive resp. influence *environment*; the part of them connected with *tool* is x resp. y. x and y together form the part of the real *border* be-

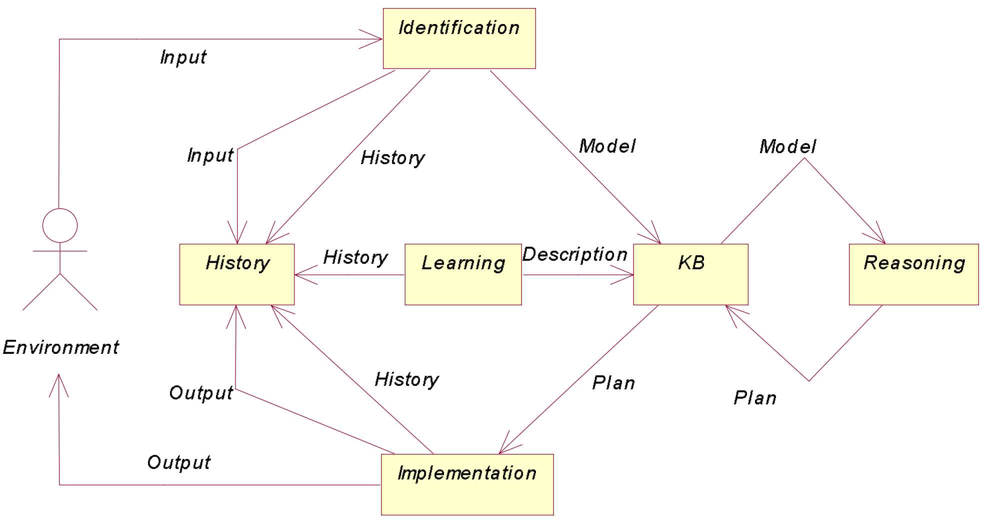

Figure 2. Problem solver

tween *PS* and *environment*, i.e. *border* between *PS* and *tool*. x' and y' form the part of *border* between *tool* and *rest of environment*. x' and y' are accessible to *PS* during *learning* but not fully accessible to *PS* later. Let us assume that *PS* can *learn* characteristics of *tool* in such a way that it can later 1) *identify tool*, 2) *model* (specify) the sequence of *values* of x' and y', if it knows the sequence of *values* of x and y. x' and y' are called *modeled inputs* and *outputs*. The equality of *modeled inputs* and *outputs*, both in *environment* and in *PS*, can be interpreted as a ***shift of border*** between *environment* and *PS*. We can interpret e.g. ***teacher***, ***society***, ***communication*** with similar *PS'*s, and an ***approach to learning*** as *tools*.

There are approaches using e.g. genetic algorithms for self-control [17]. They offer evidence of improved efficiency. They use simple learning, in my terminology. Why not generalize this and use complex learning? It should be even more efficient. This is enabled by my *assumption*: *PS* is part of *environment*. Consequently, *PS* should control itself. It can do it with all its power. For *self-control* to be practicable, *PS* must be **homogeneous**. Therefore, it must not be necessary to design and use special control mechanism for each *PS* part. *KB* should be homogeneous; this is most important. However, some homogeneity is beneficial and possible for *cognitive functions* also.

Let us consider the possibility to implement, in the framework, two types of *descriptions*, i.e. ***description*** of *object* of *environment* and *description* of *PS* ***plan***. These two *descriptions* should have had many similar properties, e.g. they should have been general, dynamic, complex, hierarchically organized, object-oriented, might have been approximate, uncertain, can utilize *tools* etc. How might they have differentiated? The only difference is that (*sub-*)*plan* should be directly related to some (*sub-*)*goal*. There is also another viewpoint: *PS* and its parts have their behaviors. They are described - generated by *plans*. *PS* and its parts are parts of *environment*. For both, it is suitable to have common *description*. As a description of external *object* and *plan* share the majority of features, it is suitable to describe them in a common way.

4 CONCLUSION

This paper proposes a framework to improve the current level of efficiency of learning. The approach shows feasibility in a sense that, on the conceptual level, the *PS* could be designed to satisfy all challenging requirements. The efficiency of learning has been analyzed and designed on a conceptual level. This is the only possible (and necessary) **beginning** of a long-term project. My intention is to continue with a detailed analysis and design of cognitive functions and KB (knowledge base), and their implementation using a contemporary software-engineering approach, i.e. utilizing object-oriented, computer-aided software-engineering approaches and tools and using contemporary knowledge about cognitive functions.

The idea to integrate all AI concepts into one system is not new. However, integration is often preached, but seldom practiced. My **contribution** is in doing the "first (small) step" toward such integration. The framework uses many concepts, but not many new ones. The exceptions are “learn from learning”**,** shifted border, that learning can create all parts of KB, and knowledge homogenization.

In higher cognitive processes, self-reflection plays an important role. AI is a cognitive process of incremental understanding the phenomenon of intelligence. What is the state of **self-reflection in AI**? According to my assessment, the field of AI is just at the beginning of its conscious self-reflection. What do I mean by this? I mean that parts of AI community - yet still small - are just starting to use specific benefits of AI work, i.e. they are starting to interpret the results of the AI process - the knowledge produced by AI - to control the AI process itself. I have tried to use this knowledge already in this paper.

ACKNOWLEDGEMENTS

This work has been partially supported by the grant GA ČR No. 102/99/1564. I also acknowledge the help of the reviewers, my colleagues J. Grim and A. Halousková, M. Kárný, my daughters, and niece.

REFERENCES

[1] J. Bůcha, On Efficiency of Learning: A Framework and Justification, Research Report No. 1983, UTIA, Prague, 2000.

[2] J. Bůcha, On Efficiency of Learning, A Framework, Research Report No. 1969, UTIA, Prague, 1999.

[3] J. Bůcha, Integration of all Cognitive Functions: Requirements, http://www.utia.cas.cz/AS_dept/bucha/ recent.html, 1999.

[4] A. G. Perez, V. R. Benjamins, Overview of Knowledge Sharing and Reuse Components: Ontologies and Problem-Solving Methods. In [18], 1-1 – 1-15, 1999.

[5] L. Console, and O. Dressler, Model-based Diagnosis in the Real World: Lessons Learned and Challenges Remaining. In T. Dean (Ed.), *IJCAI-99, Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, Stockholm, Sweden. San Francisco, Morgan Kaufmann Publishers, 1393-1400, 1999.

[6] T. R. Gruber, Towards principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, **43**, 907–928, (1995).

[7] A. Barnaras, I. Laresgoiti and J. Corera, Building and Reusing Ontologies for Electrical Network Applications, In [19], 288-302, 1996.

[8] H.S. Pinto, A. G. Pérez and J.P. Martins, Some Issues on Ontology Integration. In [18], 7-1 - 7-12, (1999).

[9] S. Thrun, C. Faloutsos, T. Mitchell, L. Wasserman, *Automated Learning and Discovery: State-Of-The-Art and Research Topics in a Rapidly Growing Field*, CMU, Pittsburgh, 1998.

[10] M. Minsky, Steps toward Artificial Intelligence, *Proc. of IRE*, **49**, 8-30 (1961).

[11] D. B. Lenat, The Dimensions of Context-Space, CYCORP, Austin, http://www.cyc.com/, 1998.

[12] M. Veloso et al., Integrating Planning and Learning: The PRODIGY Architecture, *Journal of Experimental and Theoretical Artificial Intelligence*, **7**, no. 1, (1995).

[13] The Soar Project at Carnegie Mellon University, http://www.cs.cmu.edu/afs/cs.cmu.edu/project/soar/public/www/home-page.html, 1998.

[14] R. A. Brooks, C. Breazeal (Ferrell), R. Irie, C. Kemp, M. Marjanovic, B. Scassellat and M. Williamson, Alternate Essences of Intelligence, http://www.ai.mit.edu/people/brooks/papers/group-AAAI-98.pdf. Also in AAAI-98, 1998.

[15] G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide*, Reading: Addison-Wesley, 1999.

[16] T. M. Mitchell, *Machine Learning*, McGraw-Hill Companies, N.Y. 1997.

[17] P. Collard and A. Gaspard, "Royal-Road" Landscapes for a Dual Genetic Algorithm. In [19], 1996.

[18] V. R. Benjamins (Ed.), *Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods: Lessons Learned and Future Trends*, Stockholm, 1999.

[19] W. Wahlster (Ed.), *ECAI 96, 12th European Conference on Artificial Intelligence*, J. Wiley & Sons, Ltd, 1996.