A Methodology to Develop AI Software in an Organization

Francesco Rago

***Abstract*—The challenges posed by the introduction of Artificial Intelligence (AI) tools such as Deep Learning (DL) impose the adoption of models and paradigms that are able to develop, enhance and enhance the skills of an organization. The adoption of such initiatives at an enterprise must be supported by the use of an appropriate methodological framework that interacts with all the different layers of the organization: processes, human and non-human resources, management. It has to ensure that the process improvement is consistent with the general objectives of the company. The design of software has to be improved considering the basic neural network:**

- **Function Approximation,**
- **Probability Estimation,**
- **Pattern Recognition,**
- **Clustering,**
- **Prediction.**

***Index Terms*—learning organization, artificial intelligence, deep learning, big data, technology integration**

I. Introduction

A great deal of investment and planning must be done to successfully and not marginally introduce AI tools in a company. When creating a business strategy, analytics and AI are only one piece-. AI is used effectively if strategy around it feeds into a larger business strategy, always taking into account the convergence of people, process and technology. This is congruent with technology evolution where IoT technology is evolving towards a larger and expanded IoE (Internet of Everything).

The software production process can be successful if an appropriate methodology interacts with all the different layers of the organization: processes, human and non-human resources management. It ensures process improvement consistent with the general objectives of the company.

II. RESEARCH RESULTS OF THIS PAPER

To face the complexity of a variety of today's ICT scenarios

AI computing promotes designing and developing applications in terms of autonomous software entities (agents). Agents are situated in contexts and they achieve their goals by interacting with others agents in terms of high-

level protocols. These features are well suited to tackle the complexity of developing software in modern scenarios.

This research proposes a software development methodology specifically suited to the AI paradigm.

AI software engineering has to be integrated with the concept of a learning organization as a tool for business success. Learning organization create knowledge which can be valued as an asset that makes organizations more productive and profitable.

III. Research status

Learning is the most natural of activities [2]. Machines begin to have such capability and many managers are still cynical about it. Learning is now a value embedded in software and helps create sustainable knowledge and make organizations more productive, profitable and humane places.

To introduce AI tools in an organization, there are interesting suggestions in Gardner's technical guide for Machine Learning [21] and the paper of Raza [23] describing integration of AI with software process.

IV. Understanding the Basics of Artificial Intelligence

A machine learns from data it receives by identifying patterns and relationships within the data itself. Once operational, it becomes an automated process with minimal intervention (though substantial influence) from its human counterparts.

A. AI Foundational building blocks

Artificial intelligence is built on the foundation of different technologies: machine learning and natural language processing are the two that recently had a disruptive success. The combination of constantly growing data and significantly increased computing power has extended machine learning to create deep learning. Increases in computing power enable computers to ingest bigger data and run bigger models with better algorithms. Machines can continuously analyze substantial amounts of information and transform what they have read into intelligent insights at a tremendous operational scale.

Building on the foundational skills of natural language processing, advances in machine learning have extended language capabilities to facilitate stronger collaboration between humans and machines.

F. Rago is with Megatris Comp. LLC, Sunnyvale, CA USA (e-mail: francesco.rago@megatris.info).

ISBN: 978-988-14047-9-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Machines can now chat with us through both text and speech while constantly learning and gaining insights through each interactive experience.

V. The Foundation of Our Methogology

Effective application of AI software implies a strategy that takes into account the convergence of people, management, process and technology.

First and foremost, humans are the most important resource an organization has. Data scientists, systems engineers, solution architects and business advisers are the key actors that understand unique factors within the data and the business value that will be derived from the application.

Management must assure cohesion between resources to ensure that new models are put in place in a timely manner. They have to guarantee an understanding of the changing of problem solving and decisional process that can impact how management works.

Processes have to capture input so the solution can be tailored for more accuracy and increased relevance to meet each business needs.

Finally technology has to be adopted and implementation sits within people and processes. Those areas have to receive a great deal of focus, consideration and leadership when designing any AI strategy.

VI. Models, Paradigms, Technologies

An adaptive company is based on the innovative transformation of products and services. AI tends to be developed and applied in new ways. Product cycles are shorter and require knowledge for further innovations. Equipping people to address these questions requires a new model of life-long learning, a learning model throughout the company's lifecycle that involves a parallel software life cycle with continuous review of recipes, historical series, models and parameters of neural networks.

A. Principles of Design

We introduce some empirical rules that can lead to the emergence of super-additive phenomena:

1. Information and intelligence must be distributed among a large number of entities like individuals, units and smart objects;

2. Control should not be superimposed, but must emerge as policies and result in interaction between peer-to-peer agents;

3. The system must be autocatalytic; must, therefore, be characterized by a positive feedback, which guarantees increasing returns;

4. The only way to create a truly complex system that works correctly is to assemble it incrementally starting from simple modules that can operate independently;

5. The system must be as heterogeneous as possible as diversity accelerates adaptation and strengthens the system;

6. It is necessary to minimize the errors made during the system testing phases;

7. There is no need to pursue the optimization of a particular function; you need to have multiple goals;

8. In the system testing phase, a state of continuous imbalance must be pursued because equilibrium means death of the whole system.

9. The laws governing large complex systems come from the bottom. Such laws are subject to a process of continuous adaptation, as a result of the interactions between the agents at the lowest levels of such systems.

A system built following previous rules doesn't behave like mechanical systems as can adapt to a vast set of stimuli (even not predetermined). It may evolve and its performance is not sensitive to component faults as a result of redundancy of system elements.

On the other hand, such system is not optimal relying on an emerging control mechanism (such as a price system in a free market economy) and inefficient, due to redundancy in the elements used. It is difficult to control. The system may evolve towards unwanted situations, and is non-linear and this means that, in the face of similar but not equal stimuli, different responses can be obtained. Consequently, it is difficult to predict the behavior of the system.

The complexity of the system, and the high number of interrelationships between the agents, makes the operation of the system at least obscure. In view of these considerations it is necessary to keep the system in close control.

B. Overview of methodology

The methodology organizes all the activities in four typical macro-phases: (Plan, Design, Build, and Operate), classifying them into Process, Organization, Resources, depending on how many different layers of the organization exist. The phases of are repeated recursively for each agent that autonomously aligns itself with the organization's goals in a never-ending process.

1) Plan

The Plan phase starts from the client's management of strategic addresses, the current organizational structure, the general criteria for technological transfer and the objectives of the intervention. The purpose of the initial activities is to analyze corporate culture and conduct a high-level process analysis. During the Plan phase, responsibilities for change are defined, such as:

- The definition of a steering committee,
- Ownership attribution to processes,
- Identification of the change team and agents of change,
- Identification of stakeholders.

During the Operate phase, the Plan and Designs are repeated repeatedly in the management hierarchy.

2) Design

In the Design phase, you go to the macro-level definition tasks of the process model. In this way it is possible to carry out the gap analysis, which determines the guidelines for subsequent activities. The decomposition of processes in sub-processes and tasks is carried out to the level of detail determined by the specific design requirements. In any case, decomposition levels are defined as process policies, goals and interfaces, leaving the agent creating his experience and

ISBN: 978-988-14047-9-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

inducing adaptive activities. If it is applicable, it can be based on standard pattern logic where process patterns are the set of activities, actions, work tasks or work products and similar related behavior followed by a software development life cycle:

Develop a high-level business description,

- Develop a Solution Overview Diagram,
- Identify Business Patterns,
- Identify Integration Patterns,
- Identify Composite Patterns,
- Identify Application Patterns,
- Identify Run-Time Patterns,
- Identify Run-Time and Product Mappings.

In each step the use of Machine Learning in business requires considerable data and computational power. Because Machine Learning applies analytics to such large amounts of data, and runs sophisticated algorithms, it typically requires high levels of computer performance and advanced data management capabilities.

In each Process Pattern the designer has to value the applicability of Machine Learning algorithms in the terms of:

- Function Approximation
- Probability Estimation
- Pattern Recognition
- Clustering
- Prediction

For each step the designer must define Data Collection and Preprocessing, the Architecture of networks, the appropriate training and validation procedure.

The paradigm to apply is the maps of maps to integrate different classical processing and Machine Learning algorithms. They are Relationship Module that operates to integrate Sensors Data. The Adaptation of the flows is biased by specific modules that have the task to influence Models.

a) Competences management

The Design includes the definition of competences needed to achieve the expected results and the appropriate training.

3) Build

The Build phase involves the realization of software and the development of knowledge starting from the big data of the company.

The training phase is of paramount importance for the correct accomplishment of tasks.

4) Operate

The last stage, called Operate, includes:

- The implementation of new processes and their monitoring,
- The implementation of new organizational structures,
- The implementation of new software layers,
- The assignment of roles,
- The implementation of a communication plan (natural language JSON, mobile push notifications, data collection & statistical analysis),
- Knowledge management activities with performance control over the service rendered.

At this stage, the PDBE phases are locally repeated depending on changes in system objectives and agent status.

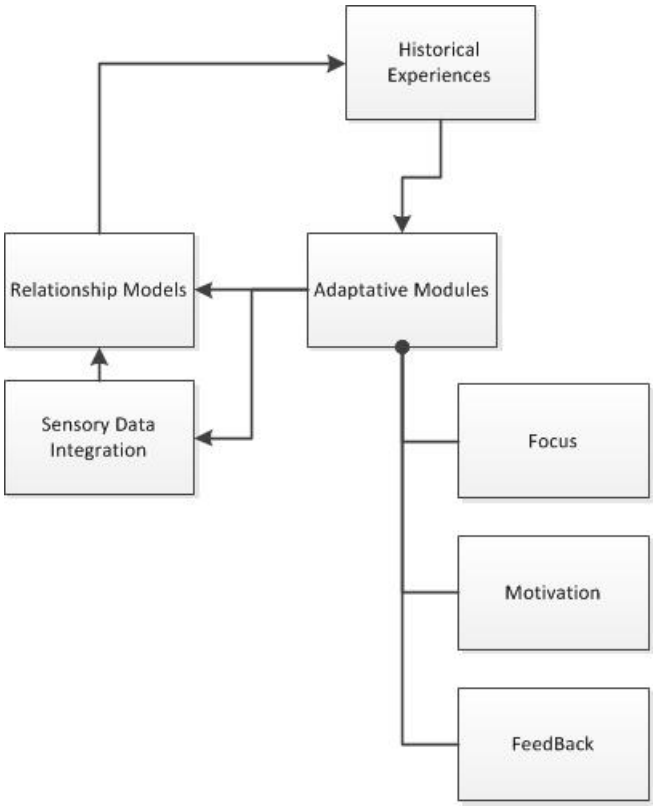

Fig. 1: Relationship Modules

VII. Conclusions for a Learning Organization

It is now accepted by most analysts that future companies will be capable of continuous learning to enjoy a strong competitive advantage. The key to economic and social progress is knowledge.

Knowledge is produced from big data in the form of a variety of trained different neural networks, but a higher competence will be required to people in the company.

Such competence development can no longer be considered a marginal activity, but must be considered as a contribution to profit creation. An investment on AI and Big Data is one of the most valuable parts of any organization, enabling competitive ability.

A strong methodology that integrates processes improvement with AI software development will enable businesses to provide fluid and adaptive knowledge. If knowledge is constantly updated together with technological evolution, this allows better prospects for the future.

References

[1] B. Garratt, (1999) "The Learning Organisation 15 years on: some personal reflections", The Learning Organization, Vol. 6 Issue: 5, pp.202-207, T.H., Davenport Prusak L., Working knowledge, Harvard Business School Press.

ISBN: 978-988-14047-9-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

[2] D. A. Garvin, Learning in Action: A Guide to Putting the Learning Organization to Work, Harvard (2000).Business School Press, Boston, MA.

[3] P. M. Senge , The Fifth Discipline: The Art & Practice of The Learning Organization Crown Publishing Group.

[4] A. Kieser, U, Koch, (2008). "Bounded rationality and organizational learning based on rule changes", Management Learning. 39 (3): 329–347.

[5] R. Albert, H. Jeong, and A. Barabasi, ‘‘Error and attack tolerance of complex networks,’’ Nature, vol. 406, pp. 378 – 382, 2000.

[6] O. Babaoglu, H. Meling, and A. Montresor, ‘‘Anthill: A framework for the development of agent based peer-to-peer systems,’’ in Proceedings of the 22nd International Conference on Distributed Computing Systems, IEEE CS Press: Vienna (A), 2002, pp. 15 – 22.

[7] L. Bass, P. Clements, and R. Kazman, Software Architectures in Practice, (2nd edn.). AddisonWesley:Reading (MA), 2003.

[8] B. Bauer, J. P. Muller, and J. Odell, ‘‘Agent UML: A formalism for specifying multiagent software systems,’’ Int. J. Software Eng. Knowl. Eng. vol. 11, no. 3, pp. 207 – 230, 2001.

[9] F. Bergenti, G. Rimassa, A. Poggi, and P. Turci, ‘‘Middleware and programming support for agent systems,’’ in Proceedings of the 2nd International Symposium from Agent Theory to Agent Implementation, Vienna (A), 2002, pp. 617 – 622.

[10] G. Booch, Object-oriented Analysis and Design, (2nd edn.), Reading (MA): Addison-Wesley, 1994.

[11] S. Bussmann, ‘‘Agent-oriented programming of manifacturing control tasks,’’ in Proceedings of the 3rd International Conference on Multi-Agent Systems, IEEE CS Press: Paris (F), 1998, pp. 57 – 63.

[12] G. Cabri, L. Leonardi, and F. Zambonelli; ‘‘Engineering mobile agent applications via contextdependent coordination,’’ IEEE Trans. Software Eng. vol. 28, no. 11, pp. 1034 – 1051, 2002.

[13] P. Ciancarini, A. Omicini, and F. Zambonelli, ‘‘Multiagent systems engineering: The coordination viewpoint,’’ in Intelligents Agents VI: Agent Theories, Architectures, and Languages, Vol. 1767 of LNAI, Springer-Verlag, 2000, pp. 250 – 259.

[14] P. Ciancarini and M. Wooldridge, ‘‘Agent-oriented software engineering,’’ in Proceedings of the 1st International Workshop on Agent-Oriented Software Engineering, Vol. 1957 of LNCS, SpringerVerlag, 2001, pp. 1 – 24.

[15] E. Carlton, Gartner: Technical professional Advice, “Preparing and Architecting for Machine Learning”, 17 January 2017 ID: G00317328,

[16] Farah Naaz Raza, “Artificial Intelligence Techniques in Software Engineering”, (AITSE) Engr., Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol I IMECS 2009, March 18 - 20, 2009, Hong Kong

ISBN: 978-988-14047-9-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)