STM JOURNALS

Journal of Artificial Intelligence Research & Advances
ISSN: 2395-6720 (Online)
Volume 4, Issue 3
www.stmjournals.com

Actors, Agents and Oracles in the Context of Artificial Intelligence

Mark Burgin*
Department of Mathematics, University of California, Los Angeles, CA, USA

Abstract
In this paper, we further develop the action theory of stratified hierarchic intelligence oriented at the creation of artificial intelligence. In this framework, we analyze and study the concept of an actor extending this concept and the corresponding model from specifically computational context to systems theory. The goal is to explore and use actor model for intelligent systems. The next step in the development of the action theory of hierarchic intelligence is investigating agents as a specific kind of actors with the main emphasis on oracles as imperative actors and empowered agents in intelligent interactions.

Keywords: *Artificial Intelligence, superintelligent, distributed intelligent managed element (DIME)*

****Author for Correspondence*** E-mail: mburgin@math.ucla.edu

INTRODUCTION

It is possible to express and observe intelligence only in actions. That is why building artificial intelligence, it is necessary to take into account behavioral issues of intelligent systems. This approach was realized in the action theory of stratified hierarchic intelligence [1–3].

As the human brain demonstrates, it is necessary to build a sophisticated complex system to achieve the level of intelligence comparable with the intelligence of people. There are different ways and techniques for decreasing complexity. Among the most efficient methods of doing this, we find stratification and hierarchical structuration of systems and processes. As a result, it is natural to suppose that an effective way to construct artificial intelligence is to use stratification and hierarchical structuration. This assumption implies necessity of a resourceful theory of stratified hierarchic intelligence, which is the main topic of this paper.

In the next section, going after introduction, a stratification of intelligence based on the Existential Triad and the Triarchic Theory of Intelligence is composed. As intelligent system has to be active in a broad sense, in the section below, we explain and further develop the actor model brought forth in computer science.

Now it is possible to treat all artificial systems, including computer systems, software systems and network systems, as agents of their users. That is why we consider agent technology in a separate section, which has become so popular in software development and networking. It is also possible to use agent technology as an efficient tool for stratification in the construction of artificial intelligence.

However, it is not clear what will happen when artificial intelligence will supersede human intelligence becoming superintelligence having an intellect that far exceeds that of humans [4,5]. Will a superintelligent artificial system still be an agent of humans or it will act independently? Will functioning of a superintelligent artificial system be safe for humans? To answer these and other questions related to superintelligence, it is necessary to be in full control of the structure and functioning of intelligent artificial system. This brings us to the concept of an oracle as a role and a component of a superintelligent artificial system.

A section is dedicated to hierarchical approach in artificial intelligence based on the mathematical theory of oracles [6,7]. This approach includes Oracle AI [8] as a special case providing new tools for exploration of artificial intelligence in general and Oracle artificial intelligence in particular.

STRATIFICATION OF INTELLIGENCE

It is workable to consider intelligence in three strata – in mentality; in the structural realm [9], for example, linguistically expressed by (oral and/or written and/or printed) words and texts in the symbolosphere [10]; and in actions and interactions in the physical world. In mentality, intelligence appears as intelligent thinking and comprehension. In the symbolosphere, intelligence comes out as intelligent speaking and writing. In the physical world, intelligence manifests itself as intelligent behavior.

This stratification corresponds to the Existential Triad [8] and forms the *locality structure* of intelligence reflecting its three principal areas:

- *Mental intelligence* mirrors the level of thinking
- *Structural* (in particular, linguistic) *intelligence* reveals the quality of expression
- *Action intelligence* exposes the worth of behavior

The locality structure of intelligence is complementary to the Triarchic Theory of Intelligence developed by Robert Sternberg [11,12]. According to Sternberg, intelligence of a human being is a mental activity aimed at purposeful adaptation to, selection and shaping of, real-world environments relevant to one's life. It means that an intelligent individual strives for adaptation to and improvement of environmental conditions in her/his life. The Triarchic Theory of Intelligence has three parts:

- *componential sub-theory* treats in the perspective of analytical giftedness, which is an ability to analyze problems and find an advantageous solutions
- *experiential sub-theory* is concerned with two aspects of experience – novelty when a person can behave efficiently in a novel situation and automation when experience allows automation of processes running them in parallel with other processes
- *practical sub-theory* reflects how an individual deal with the environment

Each sub-theory corresponds to the specific component of intelligence:

- componential sub-theory is connected to the *analytical component* of intelligence.
- experiential sub-theory is related to the *creative* or *synthetic component* of intelligence.
- practical sub-theory is associated with the *contextual component* of intelligence.

The combination of the locality structure with the Triarchic Theory of Intelligence produces nine components of intelligence:

- The analytical component of mental intelligence
- The analytical component of structural intelligence
- The analytical component of action intelligence
- The synthetic component of mental intelligence
- The synthetic component of structural intelligence
- The synthetic component of action intelligence
- The contextual component of mental intelligence
- The contextual component of structural intelligence
- The contextual component of action intelligence

This structure defines a stratified model of intelligence in general and forms a map for building an advanced artificial intelligence, in particular.

ACTOR MODEL

The Actor Model and its theory were developed in the theory of computation to provide constructive and theoretical tool for modeling, analyzing and organizing concurrent digital computations [13,14]. Here we extend this model from the theory of computational systems and processes to general systems theory because it well works for any system comprised of interacting systems, e.g., for society, organization or a computer network.

In the computational Actor Model, computation is performed by actors, which can have information about other actors and communicate asynchronously using their addresses for sending messages. In addition, actors can make decisions about local actions, create other actors and connections between actors, and determine how to reply to received messages. All these actions are treated as events in the space of actors. Usually actors are interpreted as computing devices, operators or computational processes. Actors are described by two groups of axioms - structural axioms and operational axioms.

Structural Axioms determine that the local storage of an actor may contain addresses of other actors such that satisfy one of the following conditions:
1. The addresses were provided when the actor was created.
2. The addresses have been received in messages.
3. The addresses were installed in actors created by the given actor.

Operational Axioms determine what an actor can do. Namely, an actor can:
1. Create more actors
2. Send messages to other actors
3. Receive messages from other actors

Hewitt explains that the Actor Model is rooted in physics while other models of computation are based on mathematics and/or logic [15]. As a result, the Actor Model has many properties similar to properties of physical models, especially, in quantum physics and relativistic physics. For instance, detailed observation of the arrival order of the messages for an actor affects the results of computation and can even increase indeterminacy. Usually the behavior of an actor is exactly defined when it receives a message while at other times, it may be indeterminate.

An important feature of the Actor Model is that Actor Model includes systems that are impossible in Turing's model while the latter is a special case of the Actor Model. As Milner [16] wrote, Hewitt declared that a value, an operator on values, and a process should all be actors. Taking into account that actors can be software systems, we see that software agents are a very special but essentially important case of actors. It is possible to compare relation between actors and agents with the relation between a function and a computable, e.g., recursive, function.

The Actor Model proved being very useful for concurrent computations. However, taking the concept of an actor in all its generality, we extend the Actor Model far beyond the area of computers, computer networks and computations. The basic concept in the extended Actor Model is the concept of an actor.

Definition 1. Taking a system C of interacting systems $\{R_k\ ;\ k \in K\}$, which have the lower rank than *C*, we call the systems R_k actors and treat them as actors.

Although some authors call these systems agents (cf., for example, [17,18]), it is more reasonable to call them actors because according to the common usage, an agent is an actor who/that acts on behalf of another actor. Besides, in political science and economics, an agent is a person or entity able to make decisions and take actions on behalf of, or that impact, another person or entity called the principal [19–21].

Note that the systems R_k can have different ranks. For instance, in society, actors include separate individuals, organizations, countries, and so on. Now we come to the special form of actors, which are called agents.

AGENT METHODOLOGY
In society, people use agents as advanced tools for achieving various goals and solving diverse problems. The basic distinction between conventional tools and agents is that agents can operate more or less autonomously with respect to the principals who delegated agency to the agents. For a long time, people employed only other people, birds and animals as their agents. The advent of information processing technology facilitated building and using artificial agents. Agent-based information processing systems turn out to be one of the most vibrant and focal areas of

technological research and development in the 1990s.

In this context, many assume that an agent is a software system capable of flexible autonomous action in dynamic, changeable, open environment believing that agents represent the most imperative new paradigm for software development since object-orientated programming.

In addition, agent have become popular in a diverse range of areas of information technology, including artificial intelligence, computer networks, distributed and concurrent systems, ubiquitous computing, software engineering, human-computer interaction, telematics, data analytics, machine learning, mobile systems, control systems, ambient intelligence, computer-supported cooperative labor, decision support systems, information retrieval and management, autonomous computing, electronic commerce, robotics and so on. In the context of cloud computing, agent technologies have progressively come to the foreground. The advantage of the agent paradigm is that it provides the means, at the appropriate level of abstraction, to conceive, develop and maintain flexible adaptive systems, which efficiently function in rapidly changing, irregular, unpredictable, fuzzy environment.

It is interesting to remark that agents had been introduced, studied and utilized in theoretical computer science before they came to information processing technology. Indeed, inductive Turing machines of the first and higher orders introduced in 1983 [22] utilize agent technology, which became popular in software development in the 1990s [23,24]. Namely, an inductive Turing machine *H* of the first or higher orders contains another machine *K*, which plays the role of an agent for *H* building the memory of *H*.

Practical considerations show that the agent technology developed for inductive Turing machines provides a design metaphor, models and other tools for the further development of network architecture forming a theoretical foundation for the innovative distributed intelligent managed element (DIME) network architecture (DNA), which represents a phase transition in the progress of global networks [25,26].

There are different approaches to definitions of agents (cf., for example, [27–29]. Some include in the definition of an agent the feature of being autonomous, while others consider only intelligent agents. Here we define agents in the context of the Extended Actor Model.

Definition 2. An *agent* is an actor who/that acts on behalf of another actor that is called the *principal*.

Note that in this context an agent and/or a principal can be not only an individual system, e.g., a human being, but also a system of systems, e.g., an organization. For instance, one organization can be an agent for another organization. For efficient utilization, it is necessary to consider different properties, classes and types of agents. For instance, usually only autonomous agents are considered although there are three important types of agents:

– *Controlled agents*
– *Regulated agents*
– *Autonomous agents*

Note that an agent can be controlled not by its principal.

Proposition 1. Relation “to be an agent” is not transitive.

Sometimes it is possible that an agent has more than one principal. In the case, when one agent has two principals, say *A* and *B*, we use the term *double agent*. Usually, the term *double agent* is comprehended with a negative connotation. However, the real meaning of this situation depends on relations between, goals of and intentions of the principals *A* and *B*. All these features result in existence of different kinds of double agents.

Definition 3. If *A* is an agent of *B* and *C* is an agent of *A* and *B*, then *C* is a *transitive double agent*.

Proposition 2. If *D* is a transitive double agent of *A* and *C*, *A* is an agent of *B*, and *C* is a transitive double agent of *A* and *B*, then *D* is a transitive double agent of *B* and *C*. Another type of double agents is related to orthogonal systems.

Definition 4. Two systems *A* and *B* are *orthogonal* if, at least, one of the following conditions holds:
– systems *A* and *B* are incompatible
– systems *A* and *B* are inconsistent
– systems *A* and *B* compete with one another
– systems *A* and *B* are in war

Definition 5. If systems *A* and *B* are orthogonal and *C* is an agent of *A* and *B*, then *C* is an *orthogonal double agent*.
Note that there are implicit and explicit double agents. In the situation when *C* acts as an agent of *A* but not of *B*, while *A* is an agent of *B*, then the agent *C* is an *implicit double agent* of *A* and *B*.

There are intrinsic relations between agents and oracles – agents can be oracles and oracles can be agents. These relations are studied in the next section.

ORACLE THEORY

Very often people assume that oracles only give information. This opinion is true in general [30–32] and is also related to the role of oracles in artificial intelligence where an Oracle AI is a confined AI with no physical manipulators, with which we can only interact through text messages [8]. In the mathematical theory of oracles, it is postulated that oracles can have much wider functions [7].

It is necessary to distinguish various types of oracles as they can behave differently and have different properties. Knowledge of these properties is important for the development of high level artificial intelligence. According to the relational classification, we have the following types of oracles.
– *Independent oracles* function independently of other systems.
– Functioning of *partially dependent oracles* partially relies on other systems.
– Functioning of *dependent oracles* completely relies on other systems.

Partially dependent oracles and dependent oracles can pursue their own goals or can do some work for other systems. In the latter case, we have *agent oracles*. There are three types of agent oracles:
– *Controlled agent oracles*
– *Regulated agent oracles*
– *Autonomous agent oracles*

Control is mostly performed by prescribing actions, i.e., by instructing (telling) what to do, while regulation is typically organized by restricting actions, i.e., by telling (instructing) what not to do. Restrictions play an important role in the brain functioning [33] and this has to be reflected in artificial intelligence based on mathematical models, which incorporate not only generating rules but also excluding imperatives. An example of such models is given by grammars with prohibition [34,35].

A useful structure for AI is a hierarchy of oracles, in which oracles have ranks, which are defined in the following way. An oracle of the rank 1 is an oracle for a basic system(s). An oracle of the rank *n* is an oracle for an oracle of the rank $n – 1$. Usually, an oracle of the rank 1 is an oracle for one or several basic systems [25,26], while an oracle of the rank *n* is an oracle for one or several oracles of the rank $n – 1$. The organization of oracle interactions when an oracle supervises several systems of lower ranks is more efficient than linear ranking of oracles presented in Figure 1.

Hierarchic intelligence forms a hierarchy of oracles over the base information processing system. There are several architectural solutions for implementation of this structure. A linear hierarchy of oracles is presented in Figure 1.

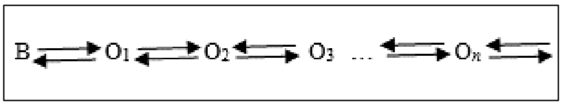

***Fig. 1:** Linear Hierarchy of Oracle.*

Here:
– B is the base information processing system
– O_1 is an oracle for B
– O_2 is an oracle for O_1
– O_3 is an oracle for O_2
…………………
– O_n is an oracle for O_{n-1}

An expanding 3-tier hierarchy of oracles is presented in Figure 2.

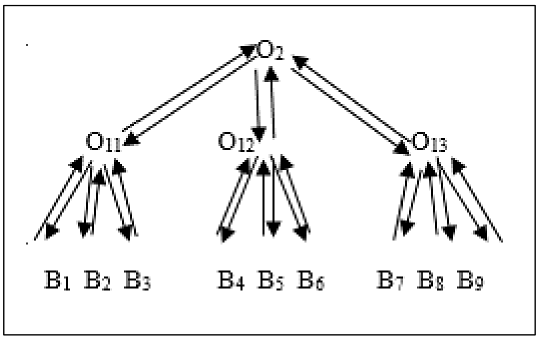

Fig. 2: *3-Tier Hierarchy of Oracles.*

Here:

- B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8, B_9 are the base information processing systems
- O_{11} is an oracle for B_1, B_2, B_3
- O_{12} is an oracle for B_4, B_5, B_6
- O_{13} is an oracle for B_7, B_8, B_9
- O_2 is an oracle for O_{11}, O_{12}, O_{13}

An expanding hierarchy of oracles allows better organization of information processing and artificial intelligence than a linear hierarchy of oracles. Different types of oracles have different properties. Let us consider transitivity of the relation of being an oracle. We remind [7] that according to the operational classification of oracles there are the following types of oracles:

1. A ***simplification oracle*** *M* for *T* is a system that solves some problems more efficiently (with less complexity) than *T*.
2. An ***augmentation oracle*** *M* for *T* is a system such that there are problems intractable for *T* but tractable for *M*.
3. An ***extension oracle*** *M* for *T* is a system such that there are problems unsolvable for *T* but solvable for *M*.

Theorems 4.1, 4.2 and 4.6 from Burgin [7] imply the following result.

Proposition 3. Relation "to be a simplification oracle" is transitive.

Theorems 4.10 and 4.12 from Burgin [7] imply the following result.

Proposition 4. Relation "to be an augmentation oracle" is transitive.

Theorem 4.21 from Burgin [7] implies the following result.

Proposition 5. Relation "to be an extension oracle" is transitive.

However, this result is true for some types of oracles.

We remind [7] that according to the functional classification of oracles there are the following types of oracles:

1. An ***informative oracle*** *M* for *T* is a system that provides some necessary information for *T*, which *T* does not have.
2. A ***performing oracle*** *M* for *T* is a system operation of which improves functioning of *T*.
3. A ***service oracle*** *M* for *T* is a system that provides services for *T* that cannot be done by *T*.

The functional classification of oracles describes functions or roles of oracles.

Proposition 6. Relation "to be an informative oracle" is not transitive.

Indeed, let us assume that a computing device *M* computes a function *f* such that a computing device *T* cannot compute and a computing device *B* computes a function *g* such that a computing device *M* cannot compute but the computing device *T* can compute all function computable by *B*. Then *M* is an informative oracle for *T* and *B* is an informative oracle for *T* but *B* is not an informative oracle for *T*.

Definition 6. A *forecasting oracle M* for *T* is a system that forecasts future of *T*.

Proposition 7. Relation "to be a forecasting oracle" is not transitive.

Indeed, let us assume that a system *M* forecasts future of *T* and a system *B* forecasts future of *M*. However, it is possible that *B* cannot forecast future of *T* and thus, *B* is not a forecasting oracle for *T* although *B* is a forecasting oracle for *M* and *M* is a forecasting oracle for *T*.

Definitions of simplification oracles employ an arbitrary complexity function *c* [7]. This allows considering specific simplification Oracles for various complexity functions, such as time complexity, space complexity and operational complexity.

Definition 7. A simplification Oracle *M* is called:

a) a *temporal Oracle* if *c* is time complexity;
b) a *space Oracle* if *c* is space complexity;
c) an *operational Oracle* if *c* is operational complexity.

Let us consider relations between different types of abstract automata from the oracle perspective in the context of artificial intelligence.

To build a high-level artificial intelligence, it would be useful to incorporate certain features of the human brain. One of the central features is that the brain learns all the time and learning is incremental [36]. Consequently, for artificial intelligence, we need to use computational models, which reflect incremental learning without stopping. These features are inherent in reflexive Turing machines, inductive Turing machines and limit Turing machines [37].

In his talk at the International Congress of mathematicians in 1958, Kleene formulated a conjecture that it might be possible that algorithms that change their programs while computing would be more powerful than Turing machines [38]. Such algorithms were formalized by building the mathematical model called *reflexive Turing machine*. However, it was proved in by Rees [39] that the class of reflexive Turing machines is equivalent to the class of Turing machines, i.e., these machines have the same computing power. This result disproved Kleene's conjecture. At the same time, it was also proved by Rees [39] that reflexive Turing machines are more efficient than Turing machines. Namely, a reflexive Turing machine can outperform any Turing machine that computes the same function [39]. The latter assertion gives us the following result.

Proposition 8. For any Turing machine T, there is a reflexive Turing machine R, which is a temporal oracle for T.

In essence, a reflexive Turing machine learns continuously and its learning is incremental similar to the processes in the brain. Moreover, a reflexive Turing machine acquires operational knowledge, which allows improving its functioning without stopping the computational process [39].

Another important mathematical model of computation is called an agent machine.

Definition 8. An *agent machine* is an automaton with two components – the basic machine and the agent. There are different types of agent machines and their agents:

- A ***memory agent*** prepares memory, e.g., establishes links between memory cells or adds new cells to the memory, for the basic machine.
- A ***data agent*** prepares data for the basic machine performing preprocessing of data.
- A ***software agent*** prepares the program, e.g., changes instructions or adds new instructions, for the basic machine.

Example 1. Inductive Turing machines of the second and higher orders have memory agents [37].

Example 2. Reflexive Turing machines can organize their functioning with software agents [39].

Example 3. The nervous system in the human eye is a data agent. Indeed, information from visual receptors (rods and cones) is not directly transmitted to the brain but at first, it is preprocessed by three kinds of neurons (bipolar cells, horizontal cells, and amacrine cells) in the retina, which play the role of data agents in the first stratum. Then it is preprocessed by ganglion cells, which plays the role of data agents in the second stratum and transmit the resulting information to the brain [40].

Definition 9. An *agent Turing machine* is an agent machine, in which both components – the basic machine and the agent machine - are Turing machines. For instance, inductive Turing machines of the first order are agent Turing machine [37].

Definition 10. An *agent inductive Turing machine* is an agent machine, in which both components – the basic machine and the agent machine - are inductive Turing machines. For instance, inductive Turing machines of the second and higher orders are agent inductive Turing machines [37].

Proposition 9. For any Turing machine T with time complexity that has growth rate higher than $O(n)$, there is an agent machine A, which is a temporal oracle for T. Proof is based on the results from [41].

CONCLUSION

The further development of the action theory of stratified hierarchic intelligence presented in this paper is oriented at the creation of artificial intelligence. We show how the generalized actor theory allows exploration of agents and oracles in a unified framework. The stratified hierarchic artificial intelligence based on oracles and agents provides a new architecture for achieving higher levels of intelligence and developing it into superintelligence. The hierarchical structure decreases complexity while utilization in higher strata more and more advanced oracles creates the base for essential growth of the power of the intelligent system.

Constructing artificial intelligence based on a hierarchy of oracles, it is important to take into account properties of oracles and agents studied in this paper. Now different people express concern with respect of potential dangers from superintelligence. Better knowledge of oracle and agent properties would also allow achieving safe and reliable superintelligence.

REFERENCES

1. Armstrong S, Sandberg A, Bostrom N. Thinking inside the Box: Controlling and Using an Oracle AI, *Mind Mach*. 2012; 22(4): 299–324p.
2. Bostrom N. *Superintelligence*: *Paths, Dangers, Strategies*, Oxford University Press, Oxford, 2014.
3. Bradshaw J. (Ed.) *Software Agents*, AAAI Press/MIT Press, Menlo Park, CA, 1997.
4. Burgin M. Inductive Turing Machines, *Notices of the Academy of Sciences of the USSR*, 1983; 270(6): 1289–1293p. (translated from Russian, v. 27, No. 3).
5. Burgin M. Knowledge in Intelligent Systems, in *Proceedings of the Conference on Intelligent Management Systems*, Varna, 1989, 281–286p.
6. Burgin M. Logical Methods in Artificial Intelligent Systems, *Vestnik of the Computer Science Society*, 1991; 2: 66–78p (in Russian).
7. Burgin M. Analogy and Argumentation in Artificial Intelligence Systems, *Vychislitelnyye Systemy* (*Logical Methods in Computer Science*), 1993; 148: 82–93p (in Russian).
8. Burgin M. Reflexive Turing Machines and Calculi, *Vychislitelnyye Systemy* (*Logical Methods in Computer Science*), 1993a; 148: 94–116, 175–176p (in Russian).
9. Burgin M. Super-recursive Algorithms as a Tool for High Performance Computing, *Proceedings of the High-Performance Computing Symposium*, San Diego, 1999, 224–228p.
10. Burgin M. *Super-recursive Algorithms*, Springer, New York/Heidelberg/Berlin, 2005.
11. Burgin M. *Grammars with Prohibition and Human-Computer Interaction*, in Proceedings of the Business and Industry Simulation Symposium, Society for Modeling and Simulation International, San Diego, California, 2005a, 143–147p.
12. Burgin M. *Structural Reality*, Nova Science Publishers, New York, 2012.
13. Burgin M. Grammars with Exclusion, *J Comput Tech Appl*. 2015; 6(2): 56–66p.
14. Burgin M. On the power of oracles in the context of hierarchical intelligence, *J Artif Intell Res Adv*. 2016; 3(2): 6–17p.
15. Burgin M. Inaccessible Information and the Mathematical Theory of Oracles, in *Information Studies and the Quest for Transdisciplinarity*: *Unity through Diversity*, World Scientific, New York/London/Singapore, 2017, 59–114p.
16. Burgin M, Dodig-Crnkovic G. *A Systematic Approach to Artificial Agents*, Preprint in Computer Science, cs.AI. 0902.3513, 2009 (electronic edition: http://arXiv.org).
17. Burgin M, Mikkilineni R. Semantic Network Organization Based on Distributed Intelligent Managed Elements: Improving Efficiency and Resiliency of Computational Processes, in *Proceedings of the Sixth International Conference on Advances in Future Internet* (AFIN 2014), Lisbon, Portugal, 2014, 1–7p.
18. Burgin M, Mikkilineni R, Morana G. Intelligent organisation of semantic networks, DIME network architecture and grid automata, *Int J Embedded Syst*. 2016; 8(4): 352–366p.
19. Burgin M, Schumann J. Three Levels of the Symbolosphere, *Semiotica*. 2006; 160(1/4): 185–202p.

20. Deacon T. *Incomplete Nature: How Mind Emerged from Matter*, W. W. Norton & Company, 2011.
21. Dinverno M, Luck M. (Eds.) *Understanding Agent Systems*, Springer Verlag, New York, 2001.
22. Doyle J. What is rational psychology? Toward a modern mental philosophy, *AI Magazine*. 1983; 4(3): 50−53p.
23. Eisenhardt KM. Agency Theory: An Assessment and Review, *Acad Manage Rev*. 1989; 14(1): 57–74p.
24. Franklin S, Graesser A. Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents, *in Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*, Springer-Verlag, 1997, 21–36p.
25. Gray P. *Psychology*, Worth Publishers, New York, 1994.
26. Curnow T. *The Oracles of the Ancient World: A Comprehensive Guide*, Duckworth, London, 2004.
27. Hawkins J. What Intelligent Machines Need to Learn from the Neocortex, *IEEE Spectrum*, June 2017, 35–39p.
28. Hewitt C. What is Commitment? Physical, Organizational, and Social, in *COIN@AAMAS'06*, 2006.
29. Hewitt C. What is computation? Actor Model versus Turing's Model, in *A Computable Universe, Understanding Computation & Exploring Nature as Computation* (H. Zenil, Ed.) World Scientific Publishing Company/Imperial College Press, 2012.
30. Hewitt C, Bishop P, Steiger R. A Universal Modular Actor Formalism for Artificial Intelligence, IJCAI'73 *Proceedings of the 3rd International Joint Conference on Artificial Intelligence*, Morgan Kaufmann Publishers Inc., San Francisco, CA, 1973, 235–245p.
31. Hibbard B. *Super-Intelligent Machines*, Kluwer Academic/Plenum Publishers, 2002.
32. Jansen J. Using Intelligent Agents to Enhance Search Engine Performance, Firstmonday, No.2/3, Available at: http://www.firstmonday.dk (1996).
33. Kleene S. Constructive and Non-Constructive Operations, in *Proceedings of the International Congress of mathematicians*, Edinburg, 1958, Cambridge University Press, 1960.
34. Leeson PT. Oracles, *Rationality and Society*, 2014; 26(2): 141–169p.
35. Mikkilineni R, Morana G, Burgin M. Oracles in Software Networks: A New Scientific and Technological Approach to Designing Self-Managing Distributed Computing Processes, Proceedings of the 2015 *European Conference on Software Architecture Workshops*, Dubrovnik/Cavtat, Croatia, ACM, 2015, 11: 1–11:8.
36. Milner R. Elements of Interaction, *Commun ACM*. 1993; 36(1): 78–89p.
37. Minsky M. *The Society of Mind*, Simon and Schuster, New York, 1986.
38. Rees R. The Theory of Principal and Agent, Pt. I, *Bull Econ Res*. 1985; 37(1): 3–26p.
39. Rees R. The Theory of Principal and Agent, Pt. II, *Bull Econ Res*. 1985; 37(2): 75–97p.
40. Sternberg RJ. *Beyond IQ: A Triarchic Theory of Intelligence*, Cambridge University Press, Cambridge, 1985.
41. Sternberg RJ. A Triarchic View of Giftedness: Theory and Practice, in (N. Coleangelo & G. A. Davis, Eds.), *Handbook of Gifted Education*, Allyn and Bacon, Boston, MA, 1997, 43–53p.

Cite this Article
Mark Burgin. Actors, Agents and Oracles in the Context of Artificial Intelligence. *Journal of Artificial Intelligence Research & Advances*. 2017; 4(3): 17–25p.