

## AGENT ORIENTED PROGRAMMING LANGUAGES: A REVIEW

S. Shoro, S. Rajper<sup>\*</sup>, N. A. Shaikh, S. Jalbani and J.-ul-F. Rajper

Department of Computer Science, Shah Abdul Latif University Khairpur, Pakistan

Received: 21 May 2018 / Accepted: 25 August 2018 / Published online: 01 September 2018

---

### ABSTRACT

Agent oriented Programming Languages are a set of instructions used to design an Agent Oriented Systems. In Agent Oriented systems, objects known as agents is an interface to accomplish particular objectives. Agents are referred as an encapsulated systems in computer which have an autonomous actions to accomplish the design goals. Agents can be self-sufficient elements, choosing their subsequent stage without the involvement of a user. The agents are particular in the capturing of structure in the natural way and the behavior of a complex system. Now a days Agents based systems are an interesting research area in computer science fields. The undertaken study aims to present the Concept, features and model of Agent Oriented Programming Languages but the main focuses of this study is on review of different programming languages used to design Agent Oriented Systems. This study will be helpful for the scholars, researchers and academicians.

**Keywords:** Agent systems, Agent Oriented Programming Languages, Multi Agent Systems, Agents.

---

Author Correspondence, e-mail: [samina.rajper@gmail.com](mailto:samina.rajper@gmail.com)

doi: <http://dx.doi.org/10.4314/jfas.v10i3.8>

### 1. INTRODUCTION

Agent-oriented Programming is devised in [1] to define an innovative paradigm of programming. It signifies a computational structure which is based on the concept of agent, shown as a



---

component of software with intellectual and communicative expertise. Agent Oriented Programming is considered as a knowledge of OOP, but AOP and OOP also have some differences [2], [1], objects directly invokes the activities of other objects whereas agents shows their aspiration for an activity to be achieved. Agent Oriented Programming Languages paradigms totally depends upon the agents. An agent is referred as a nut shelled system of computer located in a platform which is able to perform flexible and autonomous actions to design goals. [3] The concept of agents is based on Artificial Intelligence and Software Engineering. In contrast to Software Engineering Artificial Intelligence is more powerful aspect of agents. [4] The nature of agents is to be social, reactive and proactive. One similarity of agents is to have beliefs, desires and intentions.

- **Beliefs** represent its Knowledge about environment.
- **Desires** define the agent's aims,
- **Intentions** are the actions performed by the agent to affect its environment, particularly based on its beliefs.

### 1.1. Features of agents

Agents have various different features with reference to their usage and classification. The basic features of an agent are:

- **Mobility** - An agent can transfer itself from one resource to another and across different system platforms and architectures. [5]
  - **Intelligence and Size** - Agents can retain huge amounts of intelligence and can have various sizes, so they are characterized with reference to their sizes and named as big, middle sized and micro agents. It is challenging to make clear boundaries among these classifications.
    - a. A big-sized agent** Controls and occupies more than one computer. It has sufficient abilities even if it acts alone in Multi Agent Systems. A big-sized agent might be as big and as intelligent like expert system [6] in order to compete with competences for expert problem solving.
    - b. A middle-sized agent** is not beneficial without other agents in a Multi Agent Systems or without other software [7], [8], [9]. These agents are usually used in mobile agents.
    - c. Micro agents** are known as the mind agent's society. [10] They cannot hold any intelligence. Minsky follows the thought, intelligence develops in the form of results of overall activities of unintelligent and simple agents.

- **Adaptation** – Adaptive agents be able to change their behavior in changed environment. Because of adaptation agents can be robust to unpredicted changes in active environs.
- **Learning** - Agents can utilize their learning abilities for enhanced performance. Adaptive Hypermedia Systems consist of such type of agents.

## 2. RELATED WORK

Lots of investigations has been done on Agent Based Programming Languages, paradigms, tools, methodologies and Agent Oriented Software Engineering. [1] [11] [12] Describes the agent programming languages and its paradigms. [13] [14] Illustrates the agent programming in 2APL and 3APL. [15] Introduced the agents which can deal with time a certainty. [16] Discusses the various Agent Oriented Methodologies. [17] Describes various crucial engineering agent characteristics and the challenges which is being faced now a days. Whereas in [18] the author illustrates several Agent oriented tools, platforms and languages.

## 3. AGENT ORIENTED PROGRAMMING (AOP) MODEL

An AOP Model is consist of three parts:

1. A limited formal language based on clear semantics and syntax.
2. An interpreted language used to program and define the agents.
3. An "agentifier", converting neutral assets into coded agents.

## 4. AGENT ORIENTED PROGRAMMING LANGUAGES USED TO DESIGN AGENT ORIENTED SYSTEMS:

The Agent Oriented Programming language is an essential part of Agent-based Systems which provides the high level constructs and abstractions to developers to direct implementation and use of models interrelated to the agents. Still most of the agents are written in C, C++ and Java [19].\_Whereas traditional languages are not suitable for designing an agent system. But it is possible to implement those systems in languages such as C, Lisp, Pascal and Prolog [20]. Usually, OOP Smalltalk, C++ and Java, are easy to utilize in\_comprehension of agent systems and share some features with objects like as polymorphism inheritance and encapsulation etc. Whereas a number of prototype languages are been proposed to utilize in the implementation of agent based systems to provide support in comprehension of agent's distinct concepts. In this

---

undertaken study we have reviewed various programming languages on the basis of their programming paradigm and Object Oriented Programming features and found: Agent-Oriented Programming Languages, Belief desire Intention Languages (BDI), Hybrid Languages and other (generally declarative) languages are illustrated in Table-1

#### **4.1. Agent oriented programming languages**

Agent Oriented Programming Language is a set of instructions which provides high-level of concept focused on development of agents and integrates paradigms for demonstrating all structures which is defined by the model. Mostly, it allows the designers to describe agents and to bind them in particular behaviors. [21]; Shows an agent's data, holding its intellectual state; and permit agents to communicate with each other. Agent Oriented Programming paradigms were very important for the additional development of Agent Programming Languages. There are various Agent Oriented Programming Languages designed by the developers on various programming Paradigm scenarios like AGENT0, PLACA (Planning Communicating Agents), AGENT-K, MetateM, APRIL (Agent Process Interaction Language) and MAIL, VIVA, GO! [22] [1] [2] [23] [24] [25] [26] [27] [28] [29] are illustrated in Table -1

#### **4.2 Bdi-based languages and hybrid languages**

These languages are based on beliefs, desires and Intentions architecture and the Hybrid Languages are those languages which combine the AOP and BDI in to a single framework. The BDI based and Hybrid Paradigm Languages are: Agent Speak, Jason, AF-APL (Agent Factory Agent Programming Language), 3APL, 2APL, JACK Agent Language, JADDEX [30] [31] [1] [32] [33] [34] [35] [36] are illustrated in Table-1

#### **4.3 Other agent languages**

These Languages are totally based on declarative Programming Paradigm. As compared to above two these languages are more traditional but intelligent to build an intelligent agents for software. These languages are: GOAL, Golog (alGOI in LOGic), FLUX (Fluent Executer), CLAIM [37] [38] [39] [40] are illustrated in Table-1.

**Table 1.** Agent Oriented Programming Languages

<b>Agent Oriented Programming Languages Used to Design Agent Systems</b>			
<b>Programming Languages</b>	<b>Sample Languages</b>	<b>Paradigms</b>	<b>Implementation Languages</b>
<b>Agent Oriented Programming Languages</b>	AGENT0	Declarative	Prolog and Common Lisp
	PLACA	Declarative, Prototype	None
	AGENT-K	Declarative	Prolog
	MetateM	Declarative, based on linear, discrete and temporal logic	Prolog and Scheme
	APRIL	Not designed for Multi Agent Programming	C, Java, Prolog
	MAIL	Hybrid	APRIL
	VIVA	Declarative	PVM-Prolog
	GO!	Hybrid	C, java, Prolog
<b>Beliefs Desires and Intentions based Languages</b>	Agent Speak	Declarative	Jason, SIM-Talk and Agent Talk
	Jason	Hybrid	Java, Agent Talk
	AF-APL	Hybrid	Java
	3APL	Hybrid	Java, Haskell
	2APL	Hybrid	Java
	JACK	Imperative	Java
	JADEx	Hybrid	Java
<b>Other Agent Languages</b>	GOAL	Declarative	SWI-Prolog
	Golog	Declarative	Prolog
	FLUX	Declarative	Eclipse Prolog and SioStus Prolog
	CLAIM	Hybrid	Java

---

## 5. DISCUSSION/CONCLUSION

Software agents are new and emerging research area in the field of Computer Science. A number of improvements in Software agent systems development, design and implementation of Agent Systems have been made since last decades. In this study, we have illustrate the key concepts, features and model of agents but our main focus is to describe the several Agents Oriented Programming Languages used to design the Agent based Systems. The results evaluated by the literature is that Agent Programing Languages are essential in two ways, These languages can be beneficial for designing software agents and also considered as building blocks in development and deployment of composite software, typically based on agent On other side, these languages are utilized in designing and executing complex simulation modeling which holds the agent metaphor for forming and to model the critical systems. Nevertheless, these types of languages are not instantly suitable for designing real system but can be useful to understand the complex systems using simulation tools and agent-based modeling.

## 6. FUTURE WORK

In future trends we will compare the BDI based and Agent Oriented Language to find out the best one which can provide different features to design robust single agent and multi agent systems.

## 7. REFERENCES

- [1] Shoham, Y., "Agent-oriented Pogramming," *Artificial Intelligence*, vol. 60, no. 1, pp. 51-92, 1993.
- [2] Wooldridge, M.J., "Intelligent Agents, chapter 1 of Multiagent systems: a modern approach to distributed artificial intelligence", *Massachusetts Institute of Technology*, pp. 27 – 77 , 2000.
- [3] Wooldridge, M. "Agent-Based software engineering," *IEE Proc. Software Engineering* , vol. 144 (1) , pp. 26-37, 1997.
- [4] Sturm, A., Shehory, O., 2014. Agent-oriented software engineering: revisiting the state of the art. *In Agent-Oriented Software Engineering*, pp. 13-26. Springer, Berlin, Heidelberg.
- [5] Budimac, Z. ""Workflow Management System Using Mobile Agents", *Proceedings of*

- 
- ADBIS '99, Lecture Notes in Computer Science, Maribor, Slovenia*, pp. 169-178, 1999.
- [6] Huang, J., Jennings, A. R., Fox, J. "An Agent Architecture for Distributed Medical Care", *Intelligent Agents, Lecture Notes in Artificial Intelligence*, Vol 890, SpringerVerlag, pp. 219-232, 1994.
- [7] Badjonski, M., Ivanović, M., "Multi-agent System for Determination of Optimal Hybrid for Seeding", *Proceedings of EFITA '97 - First European Conference for Information Technology in Agriculture*, Copenhagen, Denmark, pp. 401-404, 1997.
- [8] Badjonski, M., Ivanović, M., Budimac, Z., "Possibility of using Multi-Agent System in Education", *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, Orlando, Florida, USA*, pp. 588-593, October 12-15-1997.
- [9] Badjonski, M., Ivanović, M., Budimac, Z., "Software Specification Using LASS", *Proceedings of Asian'97, Lecture Notes in Computer Science Vol 1345, SpringerVerlag, Kathmandu, Nepal*, pp. 375-376., 1997.
- [10] Minsky. M "The Society of Mind", *Simon and Schuster, New York* , 1986.
- [11] Ashabi. A., Khalil Salah , "Agent-Oriented Software Engineering Characteristics and Paradigm," *Journal of Multidisciplinary Engineering Science and Technology (JMEST)* , vol. 1, no. 4, pp. 176-180, November - 2014.
- [12] Michael J. Wooldridge, Nichols R. Jennings, "Agent Theories, Architectures, and Languages: A Survey".
- [13] Dastani. M, "2APL: a practical agent programming language," *Auton Agent Multi-Agent Syst*, pp. 214–248 , 2008.
- [14] KOEN V. HINDRIKS, FRANK S. DE BOER, WIEBE VAN DER HOEK and et ,al, "Agent Programming in 3APL," *Autonomous Agents and Multi-Agent Systems*, vol. 2, pp. 357-401, 1999.
- [15] Jurgen Dix, Sarit Kraus, VS Subhramanian, "Agents dealing with Time and Uncertainty," 2002.
- [16] Iglesias.C.A., Garijo, M, González, J.C., "A survey of agent-oriented methodologies.," *In International Workshop on Agent Theories, Architectures, and Languages*, pp.317-330. Springer, Berlin, Heidelberg, 1998, July.

- 
- [17] Ashabi, A. and Khalil, S., "Agent-oriented software engineering characteristics and paradigm.," *J. Multidiscip. Eng. Sci. Technol*, vol. 1, no. 4, pp.3159-0040, 2014.
- [18] Bădică, C., Budimac, Z., Burkhard, H.D. and Ivanovic, M, "Software agents: Languages, tools, platforms.," *Computer Science and Information Systems.*, vol. 8, no. 2, pp.255-298, 2011.
- [19] Weiss, G. "Agent orientation in software engineering.," *Knowledge Engineering Review*, vol. 16, no. 4, pp. 349–373, 2002.
- [20] Nwana, H., & Wooldridge, M., "Software Agent Technologies," *BT Technology Journal* , vol. 14, no. 4, pp. 68-78, 1996.
- [21] Santoro.C, "Towards an Agent Programming Language," *In 10th national workshop Towards the Future of Agent-based software systems, Parma, Italy*, 2009.
- [22] M. Badjonski, "Adaptable Java Agents – a Tool for Programming of Multi-Agent Systems.," *PhD thesis, Department of Mathematics and Informatics, Faculty of Natural Science, University of Novi Sad* , 2003.
- [23] R. Thomas, " "PLACA, an Agent Oriented Programming Language",," *PhD thesis, Computer Science Department, Stanford University, Stanford, CA 94305* , 1993.
- [24] Davies, W.H.E., Edwards, P, "Agent-K: An Integration of AOP and KQML," *In Proceedings of the Third International Conference on Information and Knowledge Management, ACM Press.*, 1994.
- [25] Barringer, H., Fisher, M., Gabbay, D., Gough, G., Owens, R, " METATEM: A Framework for Programming in Temporal Logic.," *In: Proceedings on Stepwise refinement of distributed systems: models, formalisms, correctness, REX workshop, LNCS*, vol. 430, pp. 94-129 , 1990.
- [26] Barringer, H., Fisher, M., Gabbay, D., Gough, G., Owens, R, "METATEM: An introduction," *Formal Aspects of Computing* , vol. 7, no. 5, pp. 533–549 , 1995.
- [27] McCabe, F. G., Clark, K. L., "April – Agent PProcess Interaction Language," *In Proceedings of the workshop on agent theories, architectures, and languages on Intelligent agents*, pp. 324 – 340 , 1995.
- [28] G. Wagner, "VIVA knowledge-based agent programming," *Preprint, Institut fur Informatik*,



---

Universität Leipzig, Germany., 1996.

- [29] Clark, K. L., McCabe, F. G, " Go! – A Multi-paradigm Programming Language for Implementing Multi-threaded Agents," *In Annals of Mathematics and Artificial Intelligence*, vol. 41, no. 2-4, p. 171 – 206, 2004.
- [30] Weerasooriya, D., Rao, A. S., Ramamohanarao, K, " Design of a Concurrent Agent-Oriented Language, LNAI," , Vols. Vol 890, Springer-Verlag, 1994.
- [31] Bordini, R.H., Hübner, J.F., Wooldridge, M, "Programming Multi-Agent Systems in AgentSpeak using Jason," *John Wiley & Sons*, 2007.
- [32] Hindriks, K.V., De Boer, F.S., van der Hoek, W., Meyer, J-J.C, "Agent Programming in 3APL," *Autonomous Agents and Multi-Agent Systems*, vol. 2, no. 4, pp. 357-401, 1999.
- [33] "<http://apapl.sourceforge.net/>,".
- [34] "<http://www.aosgrp.com/>,".
- [35] Pokahr, A., Brauhach, L., Lamersdorf, W, "Jadex: A BDI Reasoning Engine," pp. 149-174, Springer , 2005.
- [36] "<http://jadex-agents.informatik.uni-hamburg.de/>,".
- [37] Hindriks, K.V., De Boer, F.S., van der Hoek, W., Meyer, J-J.C, ""Agent Programming with Declarative Goals"," *Intelligent Agents VII. Agent Theories Architectures and Languages, LNCS 1986*, pp. 248-257, Springer, 2001.
- [38] J. McCarthy, " Situations, actions and causal laws.," *Technical report, Stanford University, 1963. Reprinted in Semantic Information Processing (M. Minsky ed.), MIT Press, Cambridge, Mass*, pp. 410-417 , 1968.
- [39] M. Thielscher, "Introduction to the fluent calculus," *Electronic Transactions on Artificial Intelligence*, vol. 2, no. 3-4, pp. 179–192 , 1998.
- [40] Cardelli, L., Gordon, A.D., " Mobile ambients. Foundations of Software Science and Computational Structures," *Lecture Notes in Artificial Intelligence 1378, Springer*, pp. 140-155, 1998.
- [41] Y. Shoham, "Agent-oriented programming," *Artificial Intelligence 60* , pp. 51-92 , 1993.
- [42] Nwana, H., & Wooldridge, M, "Software Agent Technologies," *BT Technology Journal*, vol. 14, no. 4, pp. pp. 68-78, 1996.

**How to cite this article:**

Shoro S, Rajper S, Shaikh NA, Jalbani S, Rajper J.-ul-F. Agent oriented programming languages: a review. *J. Fundam. Appl. Sci.*, 2018, *10(3)*, 94-103.