

Agent programming with object oriented (C++)

¹Payam Porkar Rezaeiye ²Pasha Porker Rezaeiye ³esmaeil fakhimi gheslgh mohammad beig⁴Hamed Mohseni ⁵Rona Kaviani ⁶ Mehdi Gheisari ⁷Mina Golzar

^{1, 4, 5} Department of Computer, Damavand Branch, Islamic Azad University, Damavand, Iran

²Scholl Of Computer Engineering, Iran University of Science and Technology, Tehran, Iran

³University of Eyvanekey

⁶School of Computer Science and Educational Software, Guangzhou University, Guangzhou, Guangdong Province, China, 510006

⁷ University of Qom, Qom, Faculty of Computer Engineering and IT, Iran

E-mails:

porkar@damavandiau.ac.ir p_porkar@vu.iust.ac.ir fakhimi@eyc.ac.ir

Hamedmohseni@yahoo.com RonaKaviani@yahoo.com MehdiGheisari16@gmail.com,

minaqolzar88@gmail.com

Abstract More methods and agent-oriented programming languages have been introduced based on the Java language is based, we show that C++ is an agent-oriented programming carried by the object-oriented approach, to do so, the agents are designed based on the Java programming language and language C++, in our opinion, agent structure is similar to an object and we believe that behaviors such as perception, belief and action can agent the cover object-oriented principles.

Keywords *agent-oriented, object-oriented, perception, belief, action*

2 Objects

Programming object-oriented view supports huge vast of programming language with the use of characteristic behavior and functions on data's in an abstract level. Object consists of a series of private characteristics with some numbers of functions. Access of object behavior and having a relation with it is thorough this function. Object has shown in figure 1 uses the way of data's integration and encapsulation [3].

under Java. We've paid a high price in terms of time to learn a new language. We thought to design an agent - oriented by C + +, but cannot say we have a new job. However, we prefer to have this discussion in a paper to be useful for researchers in agent field.

6 Introduction

We've used a number of popular systems, such as Prometheus [1] modeling methods for modeling multi-agent systems. The implementation would need to learn Java, as the agent-oriented programming languages such as JIAC [2] and JACK [1] were all



Fig 6 . an object

Object is an independent creature that conceal characteristics and private data's and one set of operators and functions that give them publicly to other objects that we can change the objects data's , this is a control line for object and you can see its abstract in “fig 2[4].

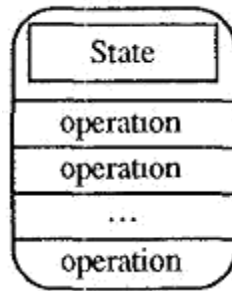


Fig2 . object

The only way to reach these characteristics and concealed data is functions or public routines. This action creates a relation for communicating with an object. That enables an object to conceal the data that don't want to show them and just showing the data that wanted [4].

2.6 Encapsulation

Object diagrams show that an object's variables make up the center, or nucleus, of the object. Methods surround and hide the object's nucleus from other objects in the program. Packaging an object's variables within the protective custody of its methods is called encapsulation [5].

2.2 Classes

Class is being used for creating an object. Variable classes introduce type of each object. Also we introduce functions and routines of each object with class [5]. A class is a kind of factory for constructing objects. The non-static parts of the class specify, or describe, what variables and methods the objects will contain. This is part of the explanation of how objects differ from classes: Objects are created and destroyed as the program runs, and there can be many objects with the same structure, if they are created using the same class [5].

3 Agents

Our society is still not considered agent final definition [6]. We can say that an agent is an independent computer system in an environment and is capable of doing an action autonomously in order to reach its goals and goals in its own environment. [6]. the first factor to consider is that the general Definition of the on agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives [6]. An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors. A robot as an agent has sensors for searching its own environment and also mechanic systems for influencing in its own environment or a software agent that its received data is a bite series and functions for influencing on environment. A generic agent is diagrammed in "Fig. 3 [7].

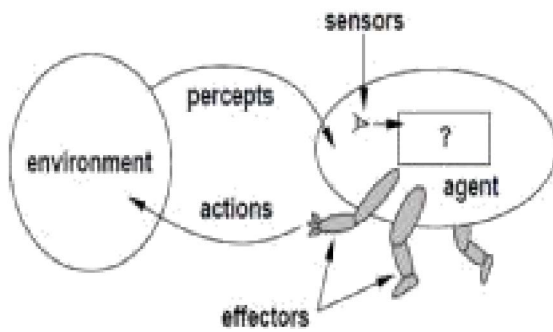


Figure 3.6. Intelligent Agents

3.6 Multi Agent Systems

Currently, the great majority of agent-based systems consist of a single agent. Yet we have to use several agents for solving complex problems, so that they can solve this problem with communicating with each other's [8]. The modern approach to artificial intelligence (AI) is centered on the concept of a rational agent. An agent that always tries to optimize an appropriate performance measure is called a rational agent. A multi agent system consists of several independent and intellectual agents that communicate with each other for solving a problem. A Multi agent system is one field of knowledge of distributed artificial intelligence [9].

4 Multi agents programming

Multi-agent systems consist of a set of autonomous and interacting computing systems called agents [11, 9, and 11]. Agents to consider independently, who are always trying to achieve their goals. Agents to perform tasks, they must be aware of their environment through sensors. Examples of such concepts and abstractions are beliefs, goals, plans, actions, events, roles, structures and communication. To develop programming in multi-agent systems analysis, modeling and programming languages are numerous. Multi-agent systems provide high-level concepts and abstractions to model and develop distributed intelligent systems. Design of multi-agent

systems as well as other debates on Software Engineering phases: requirement, specification, design, implementation and testing will follow

.Programming multi-agent systems in our need to have a general vision the details to an agent. In order, programming, agents independently of each other, set in a multi-agent system. An essential characteristic of individual agents is their autonomy.

4.6 Multi-Agent Environment

After creation of some languages for implementation of agent-oriented systems, we consider the needs of designing environments for interaction of software agents.[12]In multi agent systems, environment, is a simple part of design framework of multi agent systems that we can design it with every language that we want.[12]In most multi-agent programming frameworks the environments became simply a software component that were implemented in the same programming language as that of the interpreter of the agent-oriented programming language (e.g., Java or C++) [11, 12].

4.2 Programming Languages for Agents

One of the earliest agent-oriented programming languages is AGENT-1[13]. An agent program in AGENT-1 consists of an initial belief base, a set of capabilities, a set of commitment rules, together with a repertoire of private actions [12].The aim of these programming languages is to support the implementation of multi-agent systems, although most of them do not support the implementation of organizational abstractions, Some of these agent-oriented programming languages have an imperative programming style as they extend Java with agent concepts and abstractions, some languages have a declarative programming style as they extend logic programming languages, and yet other programming languages combine both imperative and declarative styles by integrating for example Java and Prolog [12]. Another multi-agent system is proposed programming languages is Jade (Java Agent development framework). JADE is a software framework to make easy the development of multi-agent applications in compliance with the FIPA specifications [14].FIPA does not just promote a

technology for a single application domain but a set of general technologies for different application areas that developers can integrate to make complex systems with a high degree of interoperability [14]. JADE is an Open Source project, and the complete system can be downloaded from JADE Home Page [11, 14]. And again, you see that is JADE Java based. Although we are very much agent oriented programming languages, based on Java. They do not talk about them in this paper.

4.3 Programming Languages for Multi-Agent

In the previous section we discussed was about agent independent programming. In a multi-agent system can be programmed to form correlative agents that can to work together to solve a problem, for this work [12] has suggested that we go on, we'll briefly describe them. ISLANDER One of the languages that are offered [12], In order to interpret institution specifications and execute them, a computational platform, called AMELI [11], has been developed [12]. This platform organized a mass environment, and can support the communication between agents impose rules behavioral for agents. Thing about this platform is that the agents may have violate the rules and other collective programs to ensure that the rules are. Another approach concerning specification of normative multi-agent systems by means of social and organizational concepts is MOISE⁺ [12, 11]. This modelling language can be used to specify multi-agent systems through three organizational dimensions: structural, functional. Following the MOISE⁺ Specification language, S-MOISE⁺ is an organizational middleware that provides agents access to the communication layer and the current state of the specified organization [12]. Many languages for programming multi-agent systems, which are not discussed in this paper. A programming framework for multi-agent environments is CARTAGO (Common Artifact Infrastructure for Agents Open environment) [12, 18].

1 Differences and similarities between object and agent

set of actions performed by objects in the environment are causing environmental change can be defined. Agent control over its internal state and the state itself can be said that self-object displays. The objects are active agents, where each agent has an active entity and has its own control subjects. In other words, agent independently without control by the agent or other person is able to sustain life. If the agent has the action ability and another the agent the agent wants to perform the action, the agent is autonomous act on. Object Unlike the agent does not control over his own behavior. If objects provide any other objects that makes a call to the method, if the method is public, then the objects does not have control over the implementation of method. It is clear that in most cases the object's methods should be available to other objects. Because the objects in an object-oriented system through collaboration with each other and call each other's methods, they can do their job. So object-oriented systems at the center of decision-making in conjunction with to an action method, the object is the calling object. But at the center of multi-agent systems, agents receive a request for a decision on the action to be performed. Agents that should be considered are the targeted objects, In line with agent goal-oriented its, the need for autonomy is also important, Because Considering definition target Agents, their actions are done without the involvement of other agent. Communication between objects via function calls into a predefined set of responsibilities and messaging is done. But communication from Agents by communication languages, and there is no a set of pre-defined messages or entities.

7 Design an agent in tools Prometheus

The Prometheus methodology consists of three phases [1]:

System Specification: Describe phase of system; in this phase we describe system with different goals and scenarios; connectors to environment, describing agent actions and main system characteristics [1]. Architectural design: phase of system designing architecture: in this phase the agent will be recognize, the whole structure of system will be surveyed and communication contracts will be designed [1]. Detailed design: in this phase whole details of agent will be designed in terms of agent capability, needed data and needed events. Also the whole maps and contracts between them will be considered completely [1]. We want an agent with the methodology and the Jade programming language starting designed to learn more about you with the methodology [1] visit. We are agent a robot repair work [11] that things move, repair, defense against enemy attack, we assuming that the model we have implemented. The Prometheus methodology, the agent getting the environment and its goal is doing appropriate action. Should be kept in mind, get their agent environment as a belief that the environment in their beliefs stored for use later. We are the first class to agent design, then design of a role in the repair work which we inherited from agent class.

7.6 Agent class

We need to call all classes in Java. Then of classes is agent design, we already these classes in this example [11]. Like that class belief, which describes the agent belief, Class goal which agent goal is defined as an abstract class and an entity called which message as which message is passing between Agents Or classes agent to receive and action .As you can see in “Fig 4.

```
import apltk.interpreter.data.Goal;
import apltk.interpreter.data.LogicBelief;
import apltk.interpreter.data.LogicGoal;
import apltk.interpreter.data.Message;
import eis.EnvironmentInterfaceStandard;
import eis.exceptions.ActException;
import eis.exceptions.NoEnvironmentException;
import eis.exceptions.PerceiveException;
import eis.iilang.Action;
import eis.iilang.Numerical;
import eis.iilang.Percept;
```

Fig 4. introduce the required classes

Then we'll design a class of agent called, in the form of abstract class that inherits all agent of this example. Defined class is shown in “Fig 1”. As you can see, exactly like a usual class in agent making, we have to define the variables of a particular type of classes which we've previously written, for example Class Message.

```
public abstract class Agent {
    private String name;
    private String team;
    private boolean printMind = false;
    private Set<LogicBelief> beliefs;
    protected Set<LogicGoal> goals;
    private static EnvironmentInterfaceStandard ei;
    private static Collection<String> agents;
    private static Collection<Message> messages;
    private static Map<String,String> agentsTeams;
```

Fig 5. defined class agent

As you can see in Figure 6, we define the following functions will be controlled by which agent. Like the rest of the class, we must first define the constructor function is defined, in this example the constructor function, we have parameterized constructor function .which parameter is the name of agent and its team.

```
public Agent(String name, String team) {
    this.name = name;
    this.team = team;
    if ( agents.contains(name) )
        throw new AssertionError("duplicate agent name \"" + name);
    agents.add(name);
    beliefs = new HashSet<LogicBelief>();
    goals = new HashSet<LogicGoal>();
    agentsTeams.put(name, team);
}
```

Fig1. Define agent constructor function

Other functions must be designed so that the class can be private variables that provide additional functions and elements initialized to leave factor performers. For example, messages can be designed as a queue of messages. Messages received by agent store, such as message queuing system available give it. Autonomy must also be made for private functions only in certain circumstances can agent them to execute.

8 Design an agent in C++ (Object-Oriented)

The first challenge can be that person for any goals, messages etc how they are designed. Keep in mind these are agent in the design of agreement programming constructs, supporting object-oriented programming language which can create them. For example, it can percepts a data structure that capabilities store information and presenting it in a variable to be agent. So with the help of object types in C++ can draw like, belief or messages to be then implements to class., And to enter the program, we will develop the types of variables. We just have a little intelligence and talent with a science and understanding of the correct use of languages object-oriented, is the most important agent understanding. It should be noted that the algorithm is intelligently, intelligent algorithms, this is the class of functions that we call the intelligent agent. Perhaps this is a confusing question of what duty is cooperation. Should be told cooperation on communication is built on multi-agent systems, many techniques and languages as a separate organization for cooperation has been considered. That can be used in all messaging systems of the organization, And is associated with this collaboration will be shaped.

9 Discussions and Conclusion

We first have to define the object and agent differences and similarities. My first result in this paper, eliminating the concern researchers in the field, who are fluent in an Object-oriented language and are concerned with learning a new language. Researchers should have mastery of the concept of agent and its environment. The cooperation and communicate in a method and then easily the concepts of object-oriented programming in any language can be agent. The second agent is just a name that we have put on an intelligent autonomous object and of view object-oriented the basis, a distinction was not between agent and object, my opinion, both agent object and use the object-oriented style. The difference is added to an object with

private access to a private function as autonomous and have agent name of the object.

References

- [6] Lin Padgham and Michael Winikoff, *The Prometheus Methodology*, RMIT University Melbourne, AUSTRALIA, April 2004
- [2] Rafael H. Bordini · Mehdi Dastani · Jürgen Dix, *Multi-Agent Programming*, Springer Dordrecht Heidelberg London New York Library of Congress Control Number: 2009928454 Springer Science+Business Media, LLC 2009
- [3] Apple, *Object-Oriented Programming with Objective-C*, Apple Inc. 6 Infinite Loop Cupertino, CA 95064 408-991-6060
- [4] ROGER S. CHIN AND SAMUEL T. CHANSON, *Distributed Object-Based Programming Systems*, *ACM Computing Surveys*, Vol. 23, No. 6, March 6996
- [5] David J. Eck, Durban, *Object Oriented Programming using Java*, School of Computer Science University of KwaZulu-Natal February 5, 2007
- [1] Gerhard Weiss, *Multiagent Systems A Modern Approach to Distributed Modern Approach to Artificial Intelligence*, ©6999 Massachusetts Institute of Technology ISBN 0-212-23203-0
- [7] Russell and Peter Norvig, *Artificial Intelligence*, Prentice Hall, 6995 60 G. Weiss. Multiagent systems. A modern approach to distributed artificial intelligence. The MIT Press, 6999
- [8] Katia P. Sycara, *American Association for Artificial Intelligence. All rights reserved. 0738-4102-6998*
- [9] Nikos Vlassis, *A Concise Introduction to Multi agent Systems and Distributed AI*, University of Amsterdam 2003

[60] Tristan Behrens Michael Koster Federico Schlesinger, *Multi-Agent Programming Contest*, <http://www.multiagentcontest.org/2066/> August 8, 2066

[66] Jürgen Dix Jomi Hübner M. Wooldridge. *An Introduction to MultiAgent Systems* (second edition). Wiley, 2009

[62] Mehdi Dastani, *Programming Multi-Agent Systems*, Utrecht University the Netherlands, 2066

[63] Y. Shoham. Agent-oriented programming. *Artificial Intelligence*, 10:56{92, 6993.

[64] Fabio Bellifemine⁶, Agostino Poggi², and Giovanni Rimassa, *Developing Multi-agent Systems with JADE*, C. Castelfranchi, Y. Lésperance (Eds.): *Intelligent Agents VII*, LNAI 6981, pp. 89–603, 2006.

[65] The JADE Project Home Page, 2000. Available at <http://sharon.cselt.it/projects/jade>.

[61] M. Esteva, J.A. Rodríguez-Aguilar, B. Rosell, and J.L. Arcos. AMELI: An agent-based middleware
01

for electronic institutions. In *Proceedings of the Third International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS2004)*, pages 231{243, New York, US, July 2004.

[67] J. Hübner, J.S. Sichman, and O. Boissier. Developing organised multiagent systems using the MOISE⁺ model: programming issues at the system and agent levels. *International Journal of Agent-Oriented Software Engineering*, 6(3/4):370{395, 2007

[68] A. Ricci, M. Viroli, and A. Omicini. Cartago : A framework for prototyping artifact-based environments in mas. In *E4MAS*, pages 17{81, 20

