Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 12, December 2015, pg.107 – 111

A Survey Article on Popular Artificial Intelligence Programming Languages

K.PRAMILARANI[1], M.GURUPRIYA[2]

[1]Senior Assistant Professor CSE Department, New Horizon College Of Engineering, Bangalore, India

[2]Assistant Professor, CSE Department, New Horizon College Of Engineering, Bangalore, India

[1] pramiselva@yahoo.co.in; [2] priyamano89@gmail.com

Abstract- Deciding the language for a particular problem is always a great concern. In AI programming, the implementation includes part of the problem specification process and it is not like normal software engineering approach. Artificial Intelligence programs are used for exploring and building computer programs that can efficiently simulate intelligent processes especially human intelligence such as learning, understanding, reasoning and knowledge representation etc. There are variety of programming languages available for AI programming. The first practical and most widely used AI programming language is Lisp. Prolog is preferred in Europe and Japan but America's choice is LISP. Usually choosing the language depends on the problem in hand.AI applications often work on complex information. AI language allows full flexibility in defining and manipulating programs and data. Almost many PL have restriction for the range of programs that can be used, or it may require the programmer to explicitly give relevant or irrelevant details. Keeping all these in mind most popular AI languages and their features and some example programs are discussed in this paper.

Keywords: Lisp, PL, AI, prolog, knowledge representation

I.INTRODUCTION

Allen Newell and J. Clifford Shaw and Herbert Simon of Carnegie Mellon University developed their Information Processing Language (IPL)[2][4], which was a first computer language for AI programming. IPL has list which is a highly flexible data structures. A list will hold a sequence of data item. List can hold another list as an item of the lists.[9] The main features supported by lisp are the programs that could perform general problem solving, lists, associations, associative retrieval, schemas ,frames, dynamic memory allocation, data types, recursion, functions as arguments, cooperative multitasking etc.

A. Lisp

In 1960 John McCarthy, a computer scientist invented lisp at the Massachusetts Institute of Technology[9], by combining the elements of IPL with the mathematical logical system and mathematical theory called lambda calculus. The *LISP* is the short form of LISt Processor. Linked list is the data structure of Lisp .Its source code is mainly made up of lists. By means of Lisp programs ,programmers can even create new syntax and new domain specific languages which can be embedded in Lisp .Usually LISP program code is written as the expression called symbolic expressions or s-expressions with a parenthesized lists. A function in lisp can be written as a name of function first with a list of parameters inside the parenthesis.

© 2015, IJCSMC All Rights Reserved

Let us consider a function named func with two arguments can be written in lisp as (func argu1 argu2). [9]There are several dialects such as Common Lisp, Scheme and Clojure etc. Usually the s-expressions are made up of three valid objects such as atoms, lists and strings.

Any s-expression can be considered as a valid program. LISP programs can be run either on an interpreter or as a compiled code**.** The lisp interpreter will check the source code by means of Read-Evaluate-Print Loop which will first read the program code then evaluates it and finally prints the values returned by the program. For example to add 3 numbers namely 10 ,20 and 30 by means of s-expression the following code can be typed at the interpreter prompt.(+102030) .The result will be 60.The same code can also be run as a compiled code by creating a LISP source code file and typing the following code in it.

(write (+ 10 20 30)).Actually LISP uses prefix notation. i.e operators will come first followed by the operands. Here + symbol works as the function name for the addition of the numbers. Usually LISP evolution is divided into two parts: In the first step the reader program will translate the program text into lisp object or s-expressions and then implementation of the semantics of the language in terms of these objects by an evaluator program.In that step evaluator defines syntax of Lisp forms that are formed from s-expressions. In this step only the s-expression which is of the form of lisp will be determined. The evaluator works as a function considers a valid LISP form as an argument and returns a value[3]. That is the reason the LISP expressions are in parenthesis and entire expression/form is sending to the evaluator as arguments [3].Consider the following small sample code.

```
(write-line "welcome to the world of  lisp program")
(write-line "enjoy your day")
```

When it is executed the result returned would be:
welcome to the world of lisp program
Have a nice day

B. Prolog

Next popular AI language is Prolog. Prolog is a general purpose and very popular logic programming language. Prolog coupled with AI and computational linguistics. Prolog is the first and still widely used most important logic programming language .Usually Problems in Prolog are defined as facts, axioms and logical rules etc. Old or existing facts are used for deducing new facts called inference.

The logic programming language PROLOG was created by Alain Colmerauer [4]at the University of Aix-Marseille, France in the early 1970[9].Later it was first implemented in 1972 by Colmerauer with Philippe Roussel[4]. PROLOG was further developed by the logician Robert Kowalski, a member of the AI group at the University of Edinburgh[9].This language uses resolution concept. Prolog is based upon the first order logic which is the combination of mathematics, philosophy, linguistics, and computer science[9] etc..

First-order logic considers quantified variables over non-logical objects. Prolog is a declarative languages. The program logic is expressed by means of facts, rules and relations. Usually the computation of prolog will be initiated by running a *query* over the given relations. Not only this language has been used for implementing theorem proving, expert systems and natural language processing, but Prolog environments also support creating graphical user interfaces.

Prolog is suited for rule-based logical queries such as searching in a databases, voice control systems and filling templates. In Prolog, loading of code is called as *consulting*. User interaction in Prolog can be done by entering queries at the Prolog prompt?-[4]. If there is a solution then it will be printed, otherwise Prolog writes the word no at the prompt. If the query is having multiple solutions then that can be requested by using a semi-colon. Consider the following query example:

```
?- write('welcome to the world! of prolog'), nl.
```

The output of this query will be welcome to the world! of prolog! true.?-

C. Strips

The planning methods in AI have been combined and standardized within a single syntax called Planning Domain Definition Language, or PDDL[5]. This language allows researchers to exchange some benchmark problems and compare the results. PDDL includes sublanguages for STRIPS, ADL etc. The domain description consists of the definition of domain name, conditions, actions ,rules and predicates.

STRIPS is the PDDL and it is the short form of **St**anford **R**esearch **I**nstitute **P**roblem **S**olver[5].It is an automated planner developed by Richard Fikes and Nils Nilsson in 1971 at SRI International[5].STRIPS is the base for most of the languages for automated planning problem .These languages are generally known as action languages. It uses an initial state, the goal states, and a set of actions. For each action preconditions and postconditions should be specified[5]. Preconditions are the conditions regarding what is to be done before an action is performed and postconditions are the conditions to be established after the action is performed accurately.

© 2015, IJCSMC All Rights Reserved

A STRIPS instance is a [5]quadruple (P ,O ,I ,G), in which each component has the following meaning: P is the group of conditions i.e propositional variables , O is a set of operators that is actions[5] .Each operator itself is a quadruple that includes which conditions to be true, which one must be false, which one is made to be true and which one is made to be false[5].I is the initial state which is given as the set of conditions that are initially true and all others are assumed to be false[5] and G is the goal state which is given as a pair <N ,M>, that specify which conditions are true and false. Let us consider how planning will be done by means of small problem

This monkey problem is very famous AI problem. Consider A monkey is at location A in a closed cage and one rectangular box is placed at location C which is inside the cage. Inside the cage some bunch of bananas is hanging. The monkey wants the bananas that are hanging from the ceiling in location B, the monkey should move the box and climb on to it to reach the bananas. For this kind of problem, there should be goal state ,initial state and some actions that can be done to reach the goal state must be specified[5]. Consider the following planning problem.

Initial state: At(A), position(low), RectBoxAt(C), BananaAt(B)
Goal state: ToHave(Banana)
Actions:

// moving from X to Y
Move(X, Y)
Preconditions: At(X), position(low)
Postconditions: notAt(X), At(Y)
// for climbing up on the rectangularbox
ClimbUp(Location)
Preconditions: At(Location), RectBoxAt(Location),position(low)
Postconditions: position(high), not position(low)
// for climbing down from the box
ClimbDown(Location)
Preconditions:At(Location),RectBoxAt(Location), position(high)
Postconditions: position(low), not position(high)
// to move monkey and rectangularbox from X to Y
MoveBox(X, Y)
Preconditions: At(X), RectBoxAt(X), position(low)
Postconditions: RectBoxAt(Y), not RectBoxAt(X),At(Y),not At(X)
// to take the bananas
TakeBananas(Location)
Preconditions:At(Location),BananaAt(Location), position(high)
Postconditions: Have(banana)

D. Action Description Language

ADL is a Action description language .This is also used for automated planning and scheduling system. ADL is particularly used for robots. Actually it is an advancement of STRIPS language. ADL is proposed by Pednault who is a Data abstraction and modeling specialist. He has been an IBM Research Staff Member in the Data Abstraction Research Group[6], proposed this language in 1987. It is an example of an action language.

Consider the problem of transporting cargo from one airport to another airport. Consider small helicopter is used to transport the cargo for this purpose. The helicopters need to be loaded and unloaded for transporting cargo. The necessary actions would be *loading of cargo*, *unloading of cargo* and *flying the helicopter*; So this can be expressed as express *In(c,h)* and *At(x, a)* whether a cargo *C* is in an *h* and whether an object *x* is at an airport *A*. The sequence of actions defined will be:

Action (
Load (c: cargo, h: helicopter, A: Airport)
Precondition: At(c, A) ^ At(h, A)
Effect: ¬At(c, A) ^ In(c, h)
)
Action (
Unload (c: cargo, p: helicopter, A: Airport)
Precondition: In(c, h) ^ At(h, A)
Effect: At(c, A) ^ ¬In(c, h)
)
Action (
Fly (h: helicopter, from: Airport, to: Airport)
Precondition: At(h, from)

© 2015, IJCSMC All Rights Reserved

Effect: ¬At(h, from) ^ At(h, to)
)

E. Planner

Planner is a hybrid between procedural and logical languages. It gives a procedural interpretation to logical sentences where the implications are interpreted with pattern directed inference such as (P implies Q). Planner is not an acronym but it is a programming language designed by Carl Hewitt at MIT, and first published in 1969[7].Micro-Planner and Pico-Planner were implemented, and then the whole language was implemented as *Popler* by Julian Davies at the University of Edinburgh in the POP-2 programming language[7]. Derivations such as QA4, Conniver, QLISP and Ether were important tools in Artificial Intelligence research which influenced commercial developments such as KEE and ART[7].

Pattern-directed invocation will be done by either Forward chaining or Backward chaining. For example, suppose that the goal is to find out the color of a pet Tommy, given that Tommy is barking and eating bones. Consider the rule base contains the following four rules:

1.If A is barking and A eating bones - Then A is a dog
2.If A chirping and A singing - Then A is a canary
3.If A is a dog Then A is white
4.If A is a canary - Then A is yellow

Assume the following two facts: (1)Tommy barks and (2)Tommy eats bones. With forward reasoning, the inference engine can derive that Tommy is white from the following steps:(1). First base facts indicate that "Tommy is barking" and "Tommy is eating bones", the antecedent of rule number 1 is satisfied by substituting Tommy for A in the third rule, and from the base fact the inference engine[7] gives the following conclusion: Tommy is a dog.(2). Now the antecedent of rule number 3 is satisfied by substituting Tommy for A, and the inference engine will come to the conclusion that Tommy is white.

In forward chaining the inference engine will start with the data and reasons its way to the goal. But backward chaining is working in the opposite way. In the above example, rule 2 and rule 4 is not used for reaching the goal. Because here the data determines which rules are selected and used. This method of selection is called as data-driven method. But in backward chaining inference to goal-driven is used. The forward chaining approach is widely used by expert systems, such as CLIPS. One of the main advantage of forward-chaining over the backward-chaining is the inference procedure which will make the engine best suited to dynamic situations.

F. Pop-11

POP-11 is developed at the university Brimingham.POP-11is a reflective compiled programming language with the features of interpreted language. It is an evolution of POP-2.It is supports features of procedural, declarative language constructs and pattern matching concepts.[1].POP-11 is the core language of the Poplog system.[1] Clementine data-mining system is the product of POP-11. From1999 onwards this is available as open source. An online version of ELIZA using Pop-11 is available at Birmingham. David Young used Pop-11 by combining C and Fortran to develop a teaching and interactive development tools for image processing and vision which is available in the Popvision extension to Poplog.[1].consider the following code.

```
define Double(Src) -> Result;
Src*2 -> Result;
enddefine;
```

if the input is Double(123) the it prints out:** 246

G. Python

Python is a next high level scripting language widely used in AI. It is also a interpreted and very interactive and object oriented language. Usually normal English words will be used in Python .It has very few syntactical constructions compared to other languages. Like Perl and PHP, Python is interpreted at run time and its not compiled like high level languages. Python supports all the features and techniques of oops which encapsulates the code within the objects.

Wide range of applications can be developed by means of python starting from very simple programs to complicated games. Python was developed by Guido van Rossum[8
Netherland. It is derived from many language including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages. Since python is copy righted, its source code is available under General Public License [8].Its very portable and it can be run on a variety of hardware platforms with same interface on all the platform. It supports almost all kinds of databases and GUI applications.

Python provides the support to automatic garbage collection and dynamic type checking. Python can be integrated with C,C++, COM, Active X,CORBA and java and used for designing machine learning, language processing or neural networks for artificial intelligence. Python files should have.py extension. Consider the following small python code

```
#!/usr/bin/python
print "Content-type:text/html\r\n\r\n"
```

© 2015, IJCSMC All Rights Reserved

```
print '<html>'
print '<head>'
print '<title>Hello and welcome to the World of  CGI Program</title>'
print '</head>
print '<body>'
print "welcome to the world of Python programming!";
print '<h2>Hello Word! Her is our first and simple python CGI program</h2>'
print '</body>'
print '</html>'
```

II.CONCLUSIONS

Some popular AI languages with their features and sample programs are discussed in this paper. In todays world ,one particular language will not be fit for all the kinds of problem. Some language like STRIPS best suited for planning and some other like java is best for internet application etc. Recently ,Java has become popular in some fields of AI , especially in case of intelligent agent technology and data mining. Because of its automatic garbage collection and multi-threading mechanism java is very useful in internet search engines and internet applications. With more and more increase of research in the area of web intelligence a new programming concept called agent oriented programming AOP is also budding now. In AOP, objects are nothing but the agents which will interact to achieve each and every individual goal.

Agents can exist in a structure with more complexity or with very simplicity. Agents can be autonomous entities, deciding their next step without the interference of a user, or they can be controllable working as a medium between the user and another agent. In AOP the emphasis is on system design, development platforms and connectivity.

Some critical questions are ,then how the rich number of existing AI resources developed in different languages and platforms can be integrated with other resources making use of modern system development tools like CORBA (Common Object Request Broker Architecture), generic abstract data type and annotation languages like XML, and standardized agent oriented communication language like KQML (Knowledge Query and Manipulation Language). So the future AI programming might less be concerned with questions l out answers for it can be integrated with different programming paradigms under a single best suitable communication languages for i

ACKNOWLEDGEMENT

I am very grateful to my dear husband K.Selvakumar, Senior Manager, HAL and my beloved son S.Praveen kumar for their cordial and moral support .I am very much thankful to our beloved HOD Dr.Prashanth C.S.R. for his great support and valuable information for competing this work successfully.

REFERENCES

[1] https://en.wikipedia.org/wiki/POP-11
[2] http://www.britannica.com/technology/Information-Processing-Language
[3] http://www.tutorialspoint.com
[4] http://www.britannica.com/technology/Artificial-intelligence-programming-languages
[5] https://en.wikipedia.org/wiki/STRIPS
[6] https://en.wikipedia.org/wiki/Action_description_language
[7] https://en.wikipedia.org/wiki/Planner_programming_language
[8] http://www.tutorialspoint.com/python/python_overview.htm
[9] https://en.wikipedia.org
[10] *Edwin Pednault. "IBM Research Website: Pednault". Retrieved 29 March 2013.*

© 2015, IJCSMC All Rights Reserved