IOSR Journal of Computer Engineering (IOSR-JCE)
ISSN: 2278-0661, ISBN: 2278-8727, PP: 62-65
www.iosrjournals.org

Artificial and computational intelligence

Nasim Kothiwale[1],
[1]*Computer Science and Engineering,* [1]
Shivaji University, India

Manisha Kasid[2]
[2]*Computer Science and Engineering,*
Shivaji University, India

ABSTRACT : *This paper give an introduction of Artificial and computational intelligence. The artificial intelligence (AI) is that activity devoted to making machines intelligent, and intelligence is that quality that enables an entity to function appropriately and with foresight in its environment. Computational Intelligence is the study of design of intelligent agents. The goal of computational intelligence (CI) is to understand both natural and artificial systems including bioinformatics, linguistics, robotics, games, neural networks, fuzzy systems and evolutionary computation.*

Keywords – *Artificial Intelligence, software architecture, Reusability, Software engineering.*

1. INTRODUCTION

The artificial intelligence is that activity devoted to making machines intelligent, and intelligence is that quality that enables an entity to function appropriately and with foresight in its environment. According to that definition, lots of things humans, animals, and some machines are intelligent. Machines, such as smart cameras, and many animals are at the primitive end of the extended continuum along which entities with various degrees of intelligence are arrayed. At the other end are humans, who are able to reason, achieve goals, understand and generate language, perceive and respond to sensory inputs, prove mathematical theorems, play challenging games, synthesize and summarize information, create art and music, and even write histories. Because functioning appropriately and with foresight" requires so many different capabilities, depending on the environment, we actually have several continua of intelligences with no particularly sharp discontinuities in any of them. “Getting a computer to do things which, when done by people, are said to involve intelligence.” An important goal of artificial intelligence research is to devise machine to perform various tasks normally requiring human intelligence. An intelligent system should have the ability to process complex information. This information processing is not possible as whole in principle. We need some form of partial computational strategy, which is flexible that can adapt to changes in the environment and requirements. This paper introduces software architecture for specification and verification of Artificial Intelligence system incorporating conceptual and formal framework. The focus is based on reusable components. The architecture consists of four components; Task specification layer, Problem solver layer, domain layer and an adapter layer. These four components are hierarchically organized in a layered fashion. In this architecture, the overall structure is decomposed into sub components, in a layered way such that adding new layer without changing the existing layers can change the behavior of the system. Hence, an AI system can be built in an evolutionary way by combining and adapting several reusable components without changing the existing functionality of the system. The paper is organized in to two parts. The first part of the paper summarizes the various AI architectures proposed in the literature. The second part of the paper addresses the hybrid layered software architecture for Artificial Intelligence system. Based on the proposed layered architecture, various text categorization methods are implemented in the problem solver layer and the performance is also discussed.

2. SOFTWARE ARCHITECTURE AND ITS ROLES:

Software architecture of a system describes the structure, organization of components/modules and their interactions not only to satisfy the systems’ functional and non-functional requirements but also provide conceptual integrity to the overall system structure. Software architecture concerns with the structure of large software intensive systems. [2] The architectural view is an abstract view that separates the details of implementation, algorithm and data representation and concentrates on behavioral aspects and interaction among the various components. In other words, the software architecture of the system provides an abstract description of the system by exposing certain properties and hiding others.[3] Hence the software architecture plays an important function with respect to following aspects in the development of large software intensive systems.

i. **Understandability:** It helps to understand the large system by the appropriate level of abstraction. It also exposes the high-level design constraints, thereby providing way for making architectural decision.
ii. **Reusability:** Architectural designs support the reuse of large components and provide frameworks into which components can be integrated.
iii. **Construction:** An architectural description provides a blue print for the development of the system indicating the major components and the relationship among them.
iv. **Evolution:** The architectural description of the system separates the functionality from implementation, thereby permitting to manage the concerns regarding performance, reusability and prototyping in an easy way.
v. **Analysis:** The architectural description provides a new attribute for analyzing the system with respect to quality, performance, dependency etc,. Moreover analysis of architectures built with different styles can also be made to arrive at good architectural design decisions.
vi. **Management:** Successful development of software addressing specific application depends on critical selection, analysis and evaluation of software architecture.

Artificial Intelligence systems are large and complex. The more powerful way of structuring the complexity lies in architecting the system. Hence an efficient method is needed to structure and handle the complexity of these systems. A good architecture of the system will not only satisfy the functional requirements but also the key non-functional requirements of the system such as performance, reliability, portability, maintainability etc.

3. Architectures for Artificial Intelligence Systems

The architecture of a system describes its module capabilities and how these modules operate together as a whole. The Artificial Intelligence systems can be broadly classified into reactive systems, deliberative systems and interacting systems. Over the past, numerous architectures have been proposed in the literature for these systems addressing the important features of these systems. Reactive systems are built according to behavior-based paradigm. These systems have a very simple representation of world and provide tight coupling of perception and action. Deliberative systems have a symbolic representation of the world in terms of categories such as beliefs; goals or intentions and that possess logical inference mechanism to make decisions based on their world model. The interacting systems are able to coordinate their activities through communication and negotiation.

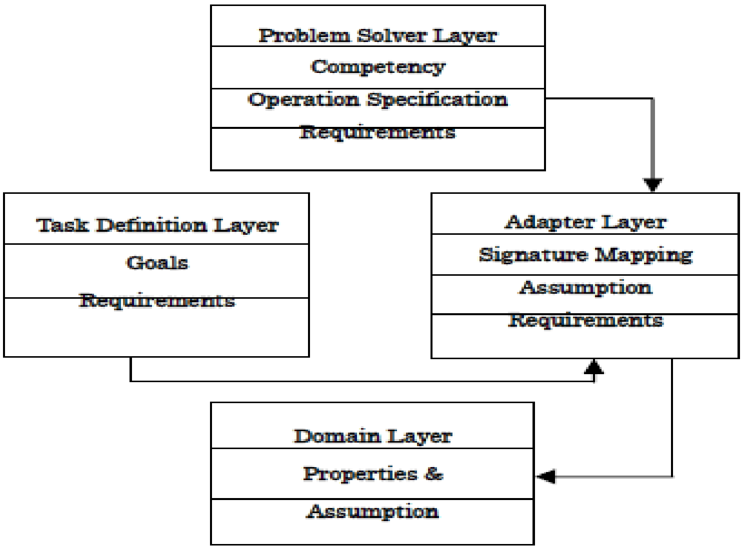
3.1 Architectures for reactive systems

These systems make decision at run time based on limited information and simple situation action rules. These architectures were often called behavior based, situated or reactive. Some researchers especially, Brooks with Subsumption architecture denied the need for symbolic representation of the world, instead the systems make their decisions based on the inputs. The decision of reactive architectures are partly guided by Simon's hypothesis which states that that the complexity of the behavior of the system can be a reflection of the complexity of the environment rather than the reflection of the system's complex internal design.

3.2 Brooks' subsumption architecture

The basic concept of subsumption architecture can be characterized as follows:

(i) Each module in the system is connected in parallel between the input and output in contrast to the conventional serial processing
(ii) Modules form layers in which higher layers can subsume lower layer functionalities and hence the name subsumption.
(iii) The lower level layers governs the basic behaviors whereas the higher level layers provide control mechanism
(iv) Augmenting a new layer at the top level without disturbing the existing layers can change the total behavior of the system.



Layered Architecture for The Artificial Intelligence System

4. Formal Specifications of Components

4.1 Task description Layer
The task description layer consists of two elements viz., the goals that should be achieved to solve a given problem. The second element of the task description layer is the definition of the requirements on the domain knowledge. Usually, axioms are used to define the requirements. The task definitions are done by algebraic specifications that provide a signature consisting of types constants functions predicates that defines the property of this component.

4.2 Problem solver layer:
The concept of problem solver layer is present in many AI frameworks. The problem solver describes the reasoning steps and which types of knowledge are needed to solve the given problem.

4.3 The Domain Layer
The description of the domain layer introduces the domain knowledge required by the problem solver and the task description layer.

5.4 The adapter Layer
Adapters are of general importance in component-based software. In the hybrid architecture, adapter layer is introduced to relate the competency of the problem solver to the functionality given by the task descriptionlayer. Further, the adapter layer introduces new requirements and assumptions because the most problems tackled with AI systems are in general complex and intractable.

5. Results and discussion:

The performance of various algorithms implemented in the problem solver layer is measured and the results obtained have been compared. The performance measurement for text classification is usually done by macro averaged recall/precision break-even point measure (F measure). In this scheme precision and recall are two important parameters used in calculating the F measure. For a category I, the precision is defined as the ratio of number of documents classified according category I by the classifier. Recall is defined as the ratio between the number of documents correctly classified to category I to the total number of documents actually belonging to I. Macro averaged F= Σ F (I)/ m Where m is the total number of categories and F(I)= (2* precision(I)*recall(I))/(recall(I)+precision (I)) The fig. 2 shows the performance of the problem solver implemented with Generalized Instance set(enhanced K-nearest neighbor) and K-nearest neighbor algorithms.
The recall level at which K-nearest neighbor starts is lower than the recall level of Generalized Instance set. For the recall level between 0.5 and 1 it is seen that the precision level of Generalized Instance set method is less than k- nearest neighbor. This is because k nearest neighbor method operates well on this data without noise. Hence, there is a slight increase in the performance for this recall level. But when the macro averaged precision

F is compared it is inferred that the average precision of GIS is more than that of K-nearest neighbor and also the macro averaged break-even point measure for GIS is better than K nearest neighbor. To over come this problem feature set pertaining to each category can be extracted and generalized instance can then be formed. When this is done GIS method will show better performance for all recall levels.

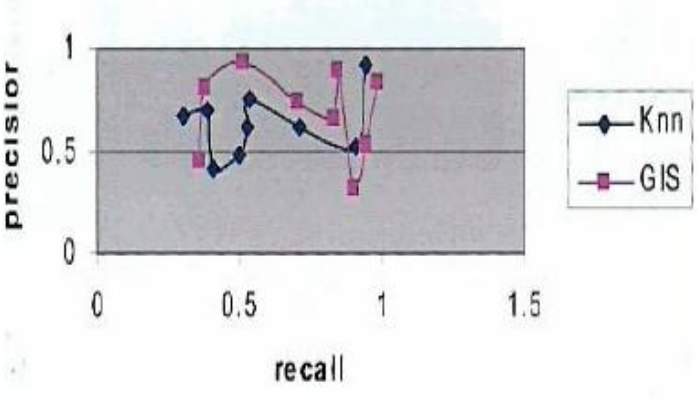


Figure 1. Relative performances of GIS and K-Nearest Neighbor Algorithm

The fig. 3 shows the performance of the problem solver implemented with enhanced K-nearest neighbor method and Bayesian method. Except for recall level at 0.4, the precision is more for enhanced K-nearest neighbor method than that of Bayesian. The precision is more for all other recall levels. The macro-averaged break-even point measure is also higher for GIS method.

6.REFERENCES

Journal Papers:

[1] McCarthy, J., Minsky, M. L., Rochester, N. and Shannon, C. E. 1955. *A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence.*

[2] Moravec, H. 1988. Mind Children: The Future of Robot and Human Intelligence. Cambridge: Harvard University Press.

[3] Russell, S. J. and Norvig, P. Artificial Intelligence: A Modern Approach. Pp. 962-964. New Jersey: Prentice Hall.

[4] J. C. Bezdek, What is computational intelligence? In: Computational Intelligence Imitating Life, pp. 1–12, IEEE Press, New York, 1994.

[5] D. Poole, A. Mackworth and R. Goebel. Computational Intelligence – A Logical Approach. Oxford University Press, New York, 1998.

[6] W. Duch, Towards comprehensive foundations of computational intelligence. In: Duch W, Mandziuk J, Eds, Challenges for Computational Intelligence. Springer 2007