AGENT ARCHITECTURE

Dr. S. Srinivasan, Mr. Amit Kumar Goel,Prof B.K Jha
Head, Department Of Computer Sciences,PDM Bhadurgarh
dss_dce@gmail.com
Research Scholar BIT-
Mesra
goelkamit@rediffmail.com
Department of Management,BITNoida, India
jha_brajendra@hotmail.com

ABSTRACT

The objective of this paper is to provide some rational, structured access to an analysis of cognitive and agent architectures.[4,5] Some familiar architectures have been considered for this preliminary analysis representing a wide range of current architectures in artificial intelligence (AI)[15]. The aim of the paper is to facilitate both an understanding of current architectures and provide insight to the development of future, improved intelligent agent architectures. The main focus is on discussing about various capabilities these architectures possess , various environments upon which these architectures act , and their memory , knowledge representation they use.

Keywords: subsumption, soar, atlantis, theo, progigy, homer, icarus, ralph-mea, multiagent system

1 INTRODUCTION

A complete functioning agent, whether biological, or simulated in software, or implemented in the form of a robot, needs an integrated collection of diverse but interrelated capabilities, i.e. architecture. At present, most work in AI and Cognitive Science addresses only components of such an architecture (e.g. vision, speech understanding, concept formation, rule learning, planning, motor control, etc.) or mechanisms and forms of representation and inference (logic engines, condition-action rules, neural nets, genetic algorithms) which might be used by many components. While such studies can make useful contributions it is important to ask, from time to time, how everything can be put together, and that requires the study of architectures.

Here we briefly discuss about some familiar architectures and their features. The study of these architectures provide a better understanding of how an intelligent agent should be . [3]

2 TYPE OF AGENT ARCHITECTURE

2.1 Subsumption Architecture

Subsumption architecture designed by Brooks - 1986 .Complicated Intelligent behavior is split into Simple behavior modules Organized into layers .Behavior is distributed rather than centralized. Response to stimuli is reflexive -- the perception-action sequence is not modulated by cognitive deliberation. The agents are organized in a bottom-up fashion. Thus, complex behaviors are fashioned from the combination of simpler, underlying ones. No explicit representation, no explicit working memory, Heuristic with in each layer as long term memory , No explicit problem solving mechanism[1] ,implicit planning by task decomposition , no learning mechanism ;Reactions to environment pre-wired into each module ,no reasoning/inference mechanism.

2.2 Gat's atlantis architecture

Integrating planning and reacting in a heterogeneous asynchronous architecture for mobile agents. It consists of Control layer, sequencing layer, deliberative layer. It is hybrid reactive/deliberative robot architecture.

2.3 Theo Architecture

The basic idea is Plan-Then-Compile. It means , integrating learning , planning and knowledge representation . It has a self Improving Problem solver .

2.4 ICARUS

Specific representation of long term memory .It uses 3 independent asynchronous modules responsible for perception , planning , effecting

2.5 Prodigy

Storing the knowledge in a form of first order predicate logic (FOPL) called Prodigy Description language (PDL). It has a modular architecture that stores the knowledge symbolically. Prodigy is an Architecture of Planning and Learning. Prodigy does not subscribe to a universal learning method like Soar but rather uses a number of different learning mechanisms. These methods allow Prodigy to be used as a test-bed for exploring relationships between problem-solving and learning [6].

2.6 Adaptive Intelligent Systems [2]

It can do reasoning and interact with other dynamic entities in real time . It has problem solving techniques. When encountering un-expected situation, it decides whether to and how to respond. It focuses attention on the most critical aspects of current situation. It will operate continuously without rebooting and also it is able to coordinate with external agent (more or less similar to human being).

2.7 Meta Reasoning

Many ideas in MAX may be traced to Prodigy. It rule-based forward – chaining engine that operates on productions. It is designed to support to modular agents. They are used to respond to a dynamic environment in a timely manner. Modules are categorized in to behavior and monitor.

2.8 Homer

It is not designed for general intelligence. The underlying philosophy is to synthesize several key areas of AI to form one complete system, like planning, learning, natural language understanding, and robotic navigation. HOMER answers questions posed by users and carries out instructions given by users. It is a modular structure. It consists of memory, planner, natural language interpreter and generator, reflective processes, plan executer.

2.9 Soar

SOAR was designed by Laird, Newell and Rosen bloom. It is the oldest and largest AI development efforts from 1983. Newell made it a developing Architectures for intelligent agents (1990). SOAR and BDI have a lot of commonalities. SOAR means STATE OPERATOR AND RESULT. Soar use all capabilities to be handled by an intelligent agent from highly routine to open-ended problems. It uses symbols (it process symbols only) . The underlying SOAR architecture is Symbolic System. SOAR follows Physical symbolic System Hypothesis (PSSH). It says that Physical symbol system is necessary and sufficient condition for general intelligence. The ultimate aim is to make a general intelligent agent.

It is based on production system. The production rule is to govern behavior. Production system uses problem space [13]. Problem space means set of all (possible) states and a set of operators. Operators transforms a particular state with in problem space to another state in the same space .It has initial stats and desired state (goal state). Operators are iteratively applied to reach goal . The sequence of steps from the initial state to goal state forms the solution or behavioral path . It uses chunking (a way of learning) and sub goaling. It uses appropriate knowledge (depends on the problem) such as procedural, declarative, episodic, semantic, iconic. It employs full range of problem solving methods. It interacts with outside world.

SOAR uses full capabilities when it tries to solve problems. Two main principle are Functionality and Performance. It provides integral approach and this helps to build general autonomous agents as well as specific intelligent agents for tasks such as natural language understanding , analogical reasoning , planning and others.

2.10 Tetlon

It is a problem solver. It uses two memory areas, short-Term memory and long-Term memory. Like human beings, interruption is allowed .It has a feature called Execution Cycle which always looks for what to do next.

2.11 Ralph-Mea

It is multiple execution architecture. Like human being, selecting best one from the environment. RALPH – MEA uses Execution Architecture (EA) to select from one state to best one. It uses the following:

1. Condition action
2. Action utility
3. Goal – based
4. Decision Theoretic

2.12 Entropy Reduction Engine

It focuses on problems that require planning , scheduling and control . It uses many different problem solving methods such as problem reduction , temporal projection , rule-based execution .

3 CAPABILITIES.

A complete functioning agent, whether biological, or simulated in software, or implemented in the form of a robot, needs an integrated collection of diverse but interrelated capabilities. While such studies can make useful contributions it is important to ask, from time to time, how everything can be put together, and that requires the study of architectures. [5]. The capabilities are broadly split as learning, planning, problem solving and reasoning.
In the following table-1A and Table-1B the rows represent the capabilities and the column corresponds to agents with specific architecture. In the cells 'Y' indicates that agent corresponds to the column possess the capability represented by the row.

Table 1: Capabilities of Agents

Capability	Subsumption	Atlantis	Theo	Prodigy	Icarus	Adaptive Intelligence System	Meta Reasoning Architecture	Homer	Soar	Tetlon	Ralph-Mea	Entropy Reduction Engine
Single Learning Method					Y			Y	Y			
Multi-Method Learning			Y	Y			Y		Y			Y
Caching			Y						Y			
Learning by Inst.				Y					Y			
Learning by Expt.				Y			Y					
Learning by Analogy				Y					Y			
Inductive Learning and Concept Acquisition			Y		Y				Y			Y
Abstarction				Y					Y			
Explanation-based learning			Y	Y			Y		Y			Y
Transfer of Learning									Y			
Planning		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Problem Solving				Y		Y		Y	Y	Y		Y
Replanning		Y			Y	Y	Y	Y	Y		Y	
Support of Multiple, Simultane-ous Goals		Y	Y			Y		Y		Y	Y	
Self Reflection			Y	Y	Y	Y	Y	Y				
Meta-Reasoning			Y	Y		Y	Y		Y		Y	
Expert System Capability						Y			Y			
Inductive and deductive Reasoning												
Prediction			Y	Y	Y	Y					Y	Y
Query Answering and providing Explanations for Decisions			Y			Y		Y				
Navigational Strategies	Y	Y	Y		Y	Y		Y				
Natural Language Understanding								Y	Y			
Learning by Perception	Y	Y	Y			Y		Y	Y			Y

Support for Inaccurate sensing		Y				Y					Y	
Robotic Tasks	Y	Y	Y			Y		Y	Y		Y	
Real-time Execution												
Focused Behavior and Processing / Selective Attentions	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	
Goal Reconstruction										Y		
Responding Intelligently to Interrupts and Failures		Y		Y	Y	Y	Y	Y	Y		Y	
Human-like Math Capability									Y			

4 ENVIRONMENT

The world itself is a very complex place for architecture agents to act in. Most architecture are designed to only deal with fractions of the total possible environmental complexity by acting in particular domains. For example, some architecture assume that the world is static and that the only things that change in the world are via an agent's actions. Other architectures may operate in dynamic environments but require that world be consistent or predictable.

An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuator [2]. A Vacuum cleaner agent is a reactive agent which means it responds when it senses dirt on the floor . Where as a Taxi Driver agent should behave in a different way since its environment is entirely different. It is a Goal directed agent .[8] It has to keep all the percepts it receives in its knowledge base .

In this section various environments with which the agents have to work are discussed. In table-2 , Rows indicate the environments and the column indicates the particular architecture . Y in the cell means that the architecture corresponding to column uses the environment represented by the row.

Table 2: Environment of Agents

	Subsumption	Atlantis	Theo	Prodigy	Icarus	Adaptive Intelligence System	Meta Reasoning Architecture	Homer	Soar	Tetlon	Ralph-Mea	Entropy Reduction Engine
Static Environment				Y					Y			
Dynamic Environment	Y	Y	Y		Y	Y	Y	Y	Y		Y	Y
Consistent Environment			Y	Y					Y			
Simulated Environment				Y			Y	Y	Y	Y	Y	
Real World Environment	Y	Y	Y		Y	Y			Y			
Complex Environment	Y	Y	Y		Y	Y	Y		Y			
Knowledge-Rich Environment			Y	Y					Y			
Input-Rich Environment						Y						
Limited Resources			Y	Y		Y		Y				
Complex Knowledge				Y	Y	Y		Y		Y		
Un-predictable	Y				Y	Y		Y	Y		Y	Y
Asynchronous	Y					Y						
Concurrent	Y	Y				Y						

Varying Priority					Y	Y	Y				Y	
Limited Response Time					Y	Y					Y	Y
Multiple Tasks						Y		Y				
Supervisor						Y			Y			

5 MEMORY, KNOWLEDGE AND KNOWLEDGE REPRESENTATION

Here the focus is to study Memory , Knowledge and Knowledge representation are handled by Agent architecture . The aim of the paper is to understand the various Knowledge Representations that agents should adapt generally . Also because of the design of the architecture of the agents that are taken in to consideration, the representations vary from one agent to another [11,12].

A Vacuum cleaner agent is a reactive agent which means , it responds when it senses dirt on the floor . Where as a Taxi Driver agent should behave in a different way since its environment is entirely different. It is a Goal directed agent. [8] it has to keep all the percepts it receives in its knowledge base . The architectures of these two typical examples should certainly differ greatly . Agent properties identify and entail the techniques and methods that were used to realize a particular architecture or architectural component. For example, most architecture includes some sort of memory. An agent is said to be a knowledge-level system [14] when it rationally brings to bear all its knowledge onto every problem it attempts to solve. Thus, knowledge is the medium of transaction at the knowledge level and the behavioral law is the principle of maximum rationality. Agent properties characterize the memory: Is the memory declarative, procedural, and episodic? Are there size limitations? Is memory uniformly accessed? Is it uniformly organized? These properties have been shown in the form a table .

In the Table 3 , Rows indicate the Memory , Knowledge and knowledge representation and the column indicates the particular architecture . Y in the cell means that the architecture corresponding to column has the type of memory or knowledge and knowledge representation .

Table 3: Memory Knowledge representation

Memory , Knowledge & Representation	Subsumption	Atlantis	Theo	Prodigy	Icarus	Adaptive Intelligence System	Meta Reasoning Architecture	Homer	Soar	Tetlon	Ralph-Mea	Entropy Reduction Engine
Forward & Back-ward Chaining				Y	Y				Y			
Impasse-driven Control			Y		Y			Y	Y	Y		
Serial Processing				Y	Y		Y	Y	Y		Y	
Parrellel Processing					Y	Y			Y		Y	Y
Asynchronous Processing		Y				Y						Y
Interruptible Processing					Y	Y		Y	Y			
Open-Loop processing					Y							Y
Closed - Loop Processing					Y							
Hierarchical Organization	Y	Y			Y	Y			Y		Y	
Modular Organization	Y			Y	Y	Y	Y	Y			Y	Y
Symbolic World Model		Y	Y	Y	Y	Y	Y	Y	Y		Y	Y
Size of the Knowledge – Base			Y			Y			Y			
Glass Box approach			Y	Y			Y	Y				
Black Box Approach		Y				Y		Y	Y		Y	
Declarative Representation				Y	Y	Y	Y				Y	Y
Procedural Representation			Y		Y				Y			Y
Global Representation			Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Uniform Access to Knowledge							Y	Y				
Knowledge Consistency			Y		Y		Y					
Homogenous (Uniform) Knowledge Representation			Y	Y	Y	Y	Y					
Hetrogenous Knowledge Representation												
No-Explicit Representation	Y											
Associate Memory												
Episodic Knowledge								Y				
Meta-Knowledge			Y				Y	Y	Y	Y		
First-Order Logic Representation				Y								
Strips like operators representation				Y			Y	Y				
Frame - Like Representation			Y				Y	Y				
Network representation					Y	Y						

6 EVALUATIONS

Based on the concepts, notation, process and pragmatics the architectures are evaluated. These are the criteria and scales for evaluating architectures. In table-4 the column represents selected architectures and rows represent the criteria considered for evaluation. Y stands for the architecture being evaluated against particular criteria representing the row . But most of the cell is blank since no proper methodologies are available. Also due to the lack of analytical criteria, the cost of demonstrations and varying specifications among different architectures, developing evaluation methods is a challenge [7, 10].

Table 4: Evaluations

Evaluation	Subsumption	Atlantis	Theo	Prodigy	Icarus	Adaptive Intelligence System	Meta Reasoning Architecture	Homer	Soar	Tetlon	Ralph-Mea	Entropy Reduction Engine
Concepts												
Internal Architecture	Y								Y			
Social architecture									Y			
Communication												
Autonomy									Y			
Pro-Activity									Y			
Distribution												
Notation												
Usability	Y											
Expressiveness												
Refinement												
Dependancy of Models												
Traceability												
Clear definition									Y			
Modularity	Y								Y			
Process									Y			
Coverage of workflows												
Management												
Complexity												
Properties of process												
Pragmatics												

Tool Support												
Connectivity												
Documentation												
Usage in Projects	Y											

CONCLUSION

The problem of AI is to describe and build agents that receive percepts from the environment and perform actions, and each such agent is implemented by a function that maps percepts to actions. It explains the role of learning as extending the reach of the designer into unknown environments, and shows how it constrains agent design, favoring explicit knowledge representation and reasoning . It analyzes basic techniques for addressing complexity . This paper has tackled the question how a developer can choose among the many development options when implementing an agent application. One key aspect here is to understand that agent technology currently oDers many problem specific solutions that address only certain types of application domains. We argue that one important foundation for making accurate choices is the availability of well-defined and comparable surveys and evaluations of artifacts such as environment and capabilities. Therefore, we have in a tabular form for evaluating many kinds of Architectures with respect to capability, environments , memory and knowledge representation . .In future work we want to employ on these tables to study Multi-Agent System Technology. The idea is to Integrate state-of-the art AI techniques into intelligent agent designs, using examples from simple, reactive agents to full knowledge-based agents with natural language capabilities and so on . This leads to the study of Multi-Agent systems and its applications. In depth analysis of various Agent architectures is to build a Multi Agent System that will be suitable for our future work on Supply Chain Management.

REFERENCES

[1] Fikes, R., Nilsson, N. (1971).strips: A new approach to the application of theorem proving to problem solving. Artificial Intelligence, 2, pp. 189-203.

[2] Firby, R. (1989). Adaptive execution in complex dynamic worlds. Ph.D. thesis. Department of Computer Science. Yale University. New Haven, Connecticut.

[3] Newell, A. and Simon, H. (1963). GPS: A program that simulates human thought. In Computers and Thought, ed. Feigenbaum and Feldman. McGraw-Hill, New York.

[4] Simon, H. (1991). Cognitive Architectures in a rational analysis: comment. In K. VanLehn (ed.), Architectures for Intelligence, pp. 25-39, Lawrence Erlbaum Associates, Hillsdale, N.J.

[5] Newell, A. (1990). Unified Theories of Cognition. Harvard University Press. Cambridge, Massachusetts.

[6] Elavine Rich , Kevin Knight and Shivshankar , Artificial Intelligence by The Mcgraw Hill Publishing Company limited .

[7] Simon, H. (1962). The architecture of complexity. Proceedings of the American Philosophical Society, 26, 467-482.

[8] Artificial Intelligence: Modern Approach by Stuart J. Russell , Peter Norvig , Prentice Hall Series in Artificial Intelligence.

[9] Kurt Vanlehn , Rule-Learning Events in the acquisition of a complex skill , Journal article by, Journal of the Learning Science ,vol 8 , 1999

[10] Philip E. Agre , Hierarchichy and History in Simon’s “Architecture of Complexity “ Journal of Learning Science 12(3) , 2003

[11] Levesque H , Brachman R , A fundamental trade off in Knowledge representation and reasoning , 1998

[12] Vincent C.Muller , Is there a future for AI without representation , springer Verlag , 2007 [13] A.S Maida , A Uniform architecture for rule-based meta-reasoning and representation :- case study . , IEEE Computer Society Press.

[14] Newell, A. (1982). The knowledge level. Artificial Intelligence. 18(1), 87-127.

[15] http://ai.eecs.umich.edu/cogarch0/