iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 **Issue:** V **Month of publication:** May 2015

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

www.ijraset.com
IC Value: 13.98

Volume 3 Issue V, May 2015
ISSN: 2321-9653

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Introduction to Artificial Intelligence

Abhishek Jain[1], Manjeet Saini[2], Manohar Kumar[3]

Department Of Computer Science, Dronacharya College of Engineering, Gurgaon

Abstract: Artificial Intelligent is simulating the human brain in real time, complete with artificial consciousness and artificial general intelligence, whereas the "mechanisms of knowledge representation in the brain which is equivalent to finding artificial intelligence". The idea of A.I. systems integration is producingunit software components, such as speech synthesizers, interoperable with other components, such as common sense knowledgebases, in order to create larger, broader and more capable A.I. systems. The main methods that have been proposed for integration are message routing, or communication protocols that the software components use to communicate with each other, often through a middleware blackboard system.

***Keywords: AI (Artificial Intelligence), Bayesian network, Ontology, Hidden Markov model, Kalman filter, Decision theory, Utility theory, Machine Learning*.**

I. INTRODUCTION

Artificial intelligence (AI) is the intelligence consists by machines or software. It is a field of study which studies the goal of creating intelligence. Major AI researchers and textbooks define this branch as "the study and design of intelligent agents", where an intelligent agent is a system that perceives its environment and takes actions that maximize its chances of success. It can also be defined as "the science and engineering of making intelligent machines".

AI research is highly technical and specialized, and is deeply divided into subfields that often fail to communicate with each other. Some of the division is due to social and cultural factors: subfields have grown up around particular institutions and the work of individual researchers. AI research is also divided by several technical issues. Some subfields focus on the solution of specific problems. Others focus on one of several possible approaches or on the use of a particular tool or towards the accomplishment of particular applications.

The central problems (or goals) of AI research include reasoning, knowledge, planning, learning, natural language processing (communication), perception and the ability to move and manipulate objects. General intelligence is still among the field's long term goals. Currently popular approaches include statistical methods, computational intelligence and traditional symbolic AI. There are a large number of tools used in AI, including versions of search and mathematical optimization, logic, methods based on probability and economics, and many others. The AI field is interdisciplinary, in which a number of sciences and professions converge, including computer science, mathematics,psychology, linguistics, philosophy and neuroscience, as well as other specialized fields such as artificial psychology.

II. INTEGRATION OF SYSTEMS

Most artificial intelligence systems involve some sort of integrated technologies, for example the integration of speech synthesis technologies with that of speech recognition. Reasons for the attention towards A.I. integration is that there have already been created a number of relatively simple A.I. systems for specific problem domains such as computer vision, speech synthesis, etc., and that integrating what's already available is a more logical approach to broader A.I. than building monolithic systems from scratch.

A. Why Integration

The focus on systems integration, especially with regard to modular approaches derive from the fact that most intelligences of sign fact scales are produced by a integration of number of individual independent working components. For example, a humanoid-type of intelligence would preferably have to be able to talk using speech synthesis, hear using speech recognition, understand using a logical (or some other undefined) mechanism, and so forth. In order to produce artificially intelligent software of broader intelligence, integration of these modalities is necessary.

B. Constructionist Design Methodology

The Constructionist design methodology is a formal methodology proposed in 2004, for use in the development of cognitive robotics, communicative humanoids and broad AI systems. The creation of such systems requires integration of a large number of functionalities that must be carefully coordinated to achieve coherent system behavior. CDM is based on iterative design steps that lead to the creation of a network of named interacting modules, communicating via explicitly typed streams and discrete messages. The OpenAIR message protocol (see below) was inspired by the CDM, and has frequently been used to aid in development of intelligent systems using CDM.

One of the first projects to use CDM was Mirage, an embodied, graphical agent visualized through augmented reality which could communicate with human users and talk about objects present in the user's physical room.

C. Tools

OpenAIR Protocol:

"OpenAIR is routing and communication protocols based on publish-subscribe architecture. It is intended to be the "glue" that allows numerous A.I. researchers to share code more effectively — "AIR to share". It is a definition or a blueprint of the "post office and mail delivery system" for distributed, multi-module systems. OpenAIR provides a core foundation upon which subsequent markup languages and semantics can be based, for e.g. gesture recognition and generation, computer vision, hardware-software interfacing etc.; for a recent example sees CVML."

OpenAIR was created to allow software components that serve their own purpose to communicate with each other to produce large scale, overall behavior of an intelligent system. A simple example would be to have a speech recognition system, and a speech synthesizer communicate with an expert system through Open AIR messages, to create a system that can hear and answer various questions through spoken dialogue.

Psyclone AIOS:

Psyclone is a software platform, or an AI operating system (AIOS), developed by Communicative Machines Laboratories for use in creating large, multi-modal A.I. systems. The system is an implementation of a blackboard system that supports the OpenAIR message protocol. Psyclone is available for free for non-commercial purposes and has therefore often been used by research institutes on low budgets and novice A.I. developers.

Elvin:

Elvin is a content-based router with a central routing station, similar to the Psyclone AIOS (see above).

OAA:

The OOA is a hybrid architecture that relies on a special inter-agent communication language (ICL) – a logic-based declarative language which is good for expressing high-level, complex tasks and natural language expressions.

CORBA:

The Common Object Request Broker Architecture (CORBA) is a standard that enables software components written in multiple computer languages and running on multiple computers to interoperate. CORBA is defined by the Object Management Group (OMG). CORBA follows similar principles as the OpenAIR protocol, and can be used for A.I. systems integration.

MOSID:

The Messaging Open Service Interface Definition (OSID) is an O.K.I. specification which provides a means of sending, subscribing and receiving messages. OSIDs are programmatic interfaces which comprise a Service Oriented Architecture for designing and building reusable and interoperable software.

III. GOALS

The general problem of simulating (or creating) intelligence has been broken down into a number of specific sub-problems. These consist of particular traits or capabilities that researchers would like an intelligent system to display. The traits described below have received the most attention:-

Deduction, reasoning, problem solving:

Early AI researchers developed algorithms that imitated the step-by-step reasoning that humans use when they solve puzzles or make logical deductions.For difficult problems, most of these algorithms can require enormous computational resources – most experience a "combinatorial explosion": the amount of memory or computer time required becomes astronomical when the problem goes beyond a certain size. The search for more efficient problem-solving algorithms is a high priority for AI research.

Human beings solve most of their problems using fast, intuitive judgments rather than the conscious, step-by-step deduction that early AI research was able to model. AI has made some progress at imitating this kind of "sub-symbolic" problem solving: embodied agent approaches emphasize the importance of sensorimotor skills to higher reasoning;neural net research attempts to simulate the structures inside the brain that give rise to this skill; statistical approaches to AI mimic the probabilistic nature of the human ability to guess.

A. Knowledge representation

Knowledge representation and knowledge engineering are central to AI research. Many of the problems machines are expected to solve will require extensive knowledge about the world. Among the things that AI needs to represent are: objects, properties, categories and relations between objects; situations, events, states and time; causes and effects; knowledge about knowledge (what we know about what other people know); and many other, less well researched domains. A representation of "what exists" is ontology: the set of objects, relations, concepts and so on that the machine knows about. The most general are called upper ontologies, which attempt to provide a foundation for all other knowledge.

Among the most difficult problems in knowledge representations are:

Default reasoning and the qualification problem:-

Many of the things people know take the form of "working assumptions." For example, if a bird comes up in conversation, people typically picture an animal that is fist sized, sings, and flies. None of these things are true about all birds. John McCarthy identified this problem in 1969 as the qualification problem: for any commonsense rule that AI researchers care to represent, there tend to be a huge number of exceptions. Almost nothing is simply true or false in the way that abstract logic requires. AI research has explored a number of solutions to this problem.

The breadth of commonsense knowledge:-

The number of atomic facts that the average person knows is astronomical. Research projects that attempt to build a complete knowledge base of commonsense knowledge(e.g., Cyc) require enormous amounts of laborious ontological engineering—they must be built, by hand, one complicated concept at a time.[52] A major goal is to have the computer understand enough concepts to be able to learn by reading from sources like the internet, and thus be able to add to its own ontology.

The subsymbolic form of some commonsense knowledge:-

Much of what people know is not represented as "facts" or "statements" that they could express verbally. For example, a chess master will avoid a particular chess position because it "feels too exposed" or an art critic can take one look at a statue and instantly realize that it is a fake. These are intuitions or tendencies that are represented in the brain non-consciously and sub-symbolically. Knowledge like this in-forms, supports and provides a context for symbolic, conscious knowledge. As with the related problem of sub-symbolic reasoning, it is hoped that situated AI, computational intelligence, or statistical AI will provide ways to represent this kind of knowledge.

B. Learning

Machine learning is the study of computer algorithms that improve automatically through experience and has been central to AI research since the field's inception.

Unsupervised learning is the ability to find patterns in a stream of input. Supervised learning includes both classification and numerical regression. Classification is used to determine what category something belongs in, after seeing a number of examples of things from several categories. Regression is the attempt to produce a function that describes the relationship between inputs and outputs and predicts how the outputs should change as the inputs change. In reinforcement learning the agent is rewarded for good responses and punished for bad ones. These can be analyzed in terms of decision theory, using concepts like utility. The

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

mathematical analysis of machine learning algorithms and their performance is a branch of theoretical computer science known as computational learning theory.

Within developmental robotics, developmental learning approaches were elaborated for lifelong cumulative acquisition of repertoires of novel skills by a robot, through autonomous self-exploration and social interaction with human teachers, and using guidance mechanisms such as active learning, maturation, motor synergies, and imitation.

C. Classifiers and statistical learning methods

The simplest AI applications can be divided into two types: classifiers ("if shiny then diamond") and controllers ("if shiny then pick up"). Controllers do however also classify conditions before inferring actions, and therefore classification forms a central part of many AI systems. Classifiers are functions that use pattern matching to determine a closest match. They can be tuned according to examples, making them very attractive for use in AI. These examples are known as observations or patterns. In supervised learning, each pattern belongs to a certain predefined class. A class can be seen as a decision that has to be made. All the observations combined with their class labels are known as a data set. When a new observation is received, that observation is classified based on previous experience. A classifier can be trained in various ways; there are many statistical and machine learning approaches. The most widely used classifiers are the neural network, kernel methods such as the support vector machine, k-nearest neighbor algorithm, Gaussian mixture model, naive Bayes classifier, and decision tree.The performances of these classifiers have been compared over a wide range of tasks. Classifier performance depends greatly on the characteristics of the data to be classified. There is no single classifier that works best on all given problems; this is also referred to as the "no free lunch" theorem. Determining a suitable classifier for a given problem is still more an art than science.

D. Logic

Logic is used for knowledge representation and problem solving, but it can be applied to other problems as well. For example, the satplan algorithm uses logic forplanning and inductive logic programming is a method for learning.

Several different forms of logic are used in AI research. Propositional or sentential logic is the logic of statements which can be true or false. First-order logic also allows the use of quantifiers and predicates, and can express facts about objects, their properties, and their relations with each other. Fuzzy logic, is a version of first-order logic which allows the truth of a statement to be represented as a value between 0 and 1, rather than simply True (1) or False (0). Fuzzy systems can be used for uncertain reasoning and have been widely used in modern industrial and consumer product control systems. Subjective logic models uncertainty in a different and more explicit manner than fuzzy-logic: a given binomial opinion satisfies belief + disbelief + uncertainty = 1 within a Beta distribution. By this method, ignorance can be distinguished from probabilistic statements that an agent makes with high confidence.

Default logics, non-monotonic logics and circumscription are forms of logic designed to help with default reasoning and the qualification problem. Several extensions of logic have been designed to handle specific domains of knowledge, such as: description logics; situation calculus, event calculus and fluent calculus (for representing events and time); causal calculus; belief calculus; and modal logics.

E. Probabilistic methods for uncertain reasoning

Many problems in AI (in reasoning, planning, learning, perception and robotics) require the agent to operate with incomplete or uncertain information. AI researchers have devised a number of powerful tools to solve these problems using methods from probability theory and economics. Bayesian networks are a very general tool that can be used for a large number of problems: reasoning (using the Bayesian inference algorithm), learning (using the expectation), planning (using decision networks) and perception (using dynamic Bayesian networks). Probabilistic algorithms can also be used for filtering, prediction, smoothing and finding explanations for streams of data, helping perception systems to analyze processes that occur over time (e.g., hidden Markov models or Kalman filters). A key concept from the science of economics is "utility": a measure of how valuable something is to an intelligent agent. Precise mathematical tools have been developed that analyze how an agent can make choices and plan, using decision theory, decision analysis, and information value theory. These tools include models such as Markov decision processes, dynamic decision networks, game theory and mechanism design.

F. Languages

AI researchers have developed several specialized languages for AI research, including Lisp and Prolog.

IV. PROBLEMS SOLVING TOOLS

AI has developed a large number of tools to solve the most difficult problems in computer science. A few of the most general of these methods are discussed below.

Search and optimization
Many problems in AI can be solved in theory by intelligently searching through many possible solutions:] Reasoning can be reduced to performing a search. For example, logical proof can be viewed as searching for a path that leads from premises to conclusions, where each step is the application of an inference rule. Planning algorithms search through trees of goals and subgoals, attempting to find a path to a target goal, a process called means-ends analysis. Robotics algorithms for moving limbs and grasping objects use local searches in configuration space. Many learning algorithms use search algorithms based on optimization.
Simple exhaustive searches are rarely sufficient for most real world problems: the search space (the number of places to search) quickly grows to astronomical numbers. The result is a search that is too slow or never completes. The solution, for many problems, is to use "heuristics" or "rules of thumb" that eliminate choices that are unlikely to lead to the goal (called "pruning the search tree"). Heuristics supply the program with a "best guess" for the path on which the solution lies. Heuristics limit the search for solutions into a smaller sample size.
A very different kind of search came to prominence in the 1990s, based on the mathematical theory of optimization. For many problems, it is possible to begin the search with some form of a guess and then refine the guess incrementally until no more refinements can be made. These algorithms can be visualized as blind hill climbing: we begin the search at a random point on the landscape, and then, by jumps or steps, we keep moving our guess uphill, until we reach the top. Other optimization algorithms are simulated annealing, beam search and random optimization.
Evolutionary computation uses a form of optimization search. For example, they may begin with a population of organisms (the guesses) and then allow them to mutate and recombine, selecting only the fittest to survive each generation (refining the guesses). Forms of evolutionary computation include swarm intelligence algorithms (such as ant colony or particle swarm optimization) and evolutionary algorithms (such as genetic algorithms, gene expression programming, and genetic programming).

A. Evaluating progress

Alan Turing proposed a general procedure to test the intelligence of an agent now known as the Turing test. This procedure allows almost all the major problems of artificial intelligence to be tested. However, it is a very difficult challenge and at presents all agents fail.
Artificial intelligence can also be evaluated on specific problems such as small problems in chemistry, hand-writing recognition and game-playing. Such tests have been termedsubject matter expert Turing tests. Smaller problems provide more achievable goals and there are an ever-increasing number of positive results.[160]

One classification for outcomes of an AI test is:
Optimal: it is not possible to perform better.
Strong super-human: performs better than all humans.
Super-human: performs better than most humans.
Sub-human: performs worse than most humans.

For example, performance at draughts (i.e. checkers) is optimal, performance at chess is super-human and nearing strong super-human (see computer chess: computers versus human) and performance at many everyday tasks (such as recognizing a face or crossing a room without bumping into something) is sub-human.
A quite different approach measures machine intelligence through tests which are developed from mathematical definitions of intelligence. Examples of these kinds of tests start in the late nineties devising intelligence tests using notions from Kolmogorov complexity and data compression.[163] Two major advantages of mathematical definitions are their applicability to nonhuman intelligences and their absence of a requirement for human testers.
A derivative of the Turing test is the Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA). As the name implies, this helps to determine that a user is an actual person and not a computer posing as a human. In contrast to the standard Turing test, CAPTCHA administered by a machine and targeted to a human as opposed to being administered by a human

and targeted to a machine. A computer asks a user to complete a simple test then generates a grade for that test. Computers are unable to solve the problem, so correct solutions are deemed to be the result of a person taking the test. A common type of CAPTCHA is the test that requires the typing of distorted letters, numbers or symbols that appear in an image undecipherable by a computer.

V. USAGE

Game AI/heuristic algorithms are used in a wide variety of quite disparate fields inside a game. The most obvious is in the control of any NPCs in the game, although scripting is currently the most common means of control. Pathfinding is another common use for AI, widely seen in real-time strategy games. Pathfinding is the method for determining how to get a NPC from one point on a map to another, taking into consideration the terrain, obstacles and possibly "fog of war". Beyond path finding, navigation is a sub-field of Game AI focusing on giving NPCs the capability to navigate in their environment, finding a path to a target while avoiding collisions with other entities (other NPC, players...) or collaborating with them (group navigation).
The concept of emergent AI has recently been explored in games such as *Creatures*, *Black & White* and *Nineteendogs* and toys such as Tamagotchi. The "pets" in these games are able to "learn" from actions taken by the player and their behavior is modified accordingly. While these choices are taken from a limited pool, it does often give the desired illusion of intelligence on the other side of the screen.
Uses of AI in games beyond NPCs
AI developments can play roles in game AI beyond the traditional paradigm of AI controlling NPC behavior. He highlights four other potential application areas:
Player-experience modeling: Discerning the ability and emotional state of the player, so as to tailor the game appropriately. This can include dynamic game difficulty balancing, which consists in adjusting the difficulty in a video game in real-time based on the player's ability. Game AI may also help deduce player intent (such as gesture recognition).
Procedural-content generation: Creating elements of the game environment like environmental conditions, levels, and even music in an automated way. AI methods can generate new content or interactive stories.
Data mining on user behavior: This allows game designers to explore how people use the game, what parts they play most, and what causes them to stop playing, allowing developers to tune gameplay or improve monetization.

Alternate approaches to NPCs: These include changing the game set-up to enhance NPC believability and exploring social rather than individual NPC behavior.

VI. CHALLENGES AND SOLUTIONS

Collaboration is an integral part of software development as evidenced by the size of software companies and size of their software departments. Among the tools to ease software collaboration are various procedures and standards that developers can follow to ensure quality, reliability and that their software is compatible with software created by others (such as W3Cstandards for webpage development). However, collaboration in fields of A.I. has been lacking, for the most part not seen outside of the respected schools, departments or research institutes (and sometimes not within them either). This presents practitioners of A.I. systems integration with a substantial problem and often causes A.I. researchers to have to reinvent the wheel' each time they want a specific functionality to work with their software. Even more damaging is the "not invented here" syndrome, which manifests itself in a strong reluctance of A.I. researchers to build on the work of others.
With the increased popularity of the free software movement, a lot of the software being created, including A.I. systems, that is available for public exploit. The next natural step is to merge these individual software components into coherent, intelligent systems of a broader nature. As a multitude of components (that often serve the same purpose) have already been created by the community, the most accessible way of integration is giving each of these components an easy way to communicate with each other. By doing so, each component by itself becomes a module which can then be tried in various settings and configurations of larger architectures.
Many online communities for A.I. developers exist where tutorials, examples and forums aim at helping both beginners and experts build intelligent systems (for example the AI Depot, Generation 5). However, few communities have succeeded in making a certain standard or a code of conduct popular to allow the large collection of miscellaneous systems to be integrated with any ease.

Recently, however, there have been focused attempts at producing standards for A.I. research collaboration, Mindmakers.org is an online community specifically created to harbor collaboration in the development of A.I. systems. The community has proposed the OpenAIR message and routing protocol for communication between software components, making it easier for individual developers to make modules instantly integrateble into other peoples' projects.

VII. APPLICATIONS OF ARTIFICIAL INTELLIGENCE

In video games, artificial intelligence is used to generate intelligent behaviors primarily in non-player characters (NPCs), often simulatinghuman-like intelligence. The techniques used typically draw upon existing methods from the field of artificial intelligence (AI). However, the term game AI is often used to refer to a broad set of algorithms that also include techniques from control theory, robotics, computer graphics and computer science in general.

Since game AI for NPCs is centered on appearance of intelligence and good gameplay within environment restrictions, its approach is very different from that of traditional AI; workarounds and cheats are acceptable and, in many cases, the computer abilities must be toned down to give human players a sense of fairness. This, for example, is true in first-person shooter games, where NPCs' otherwise perfect aiming would be beyond human skill.

For example: Creatures (1996), Sid Meier's Alpha Centauri (1999),Halo: Combat Evolved (2001),F.E.A.R. (2005), S.T.A.L.K.E.R. series (2007-), Far Cry 2 (2008).

REFERENCES

[1] Schaeffer, Jonathan. One Jump Ahead: Challenging Human Supremacy in Checkers, 1997,2009, Springer, ISBN 978-0-387-76575-4. Chapter 6.
[2] McCorduck, Pamela (2004), Machines Who Think (2nd ed.), Natick, MA: A. K. Peters, Ltd., ISBN 1-56881-205-1, p. 480-483
[3] Schwab, 2004, p. 97-112
[4] Yannakakis, Geogios N (2012). "Game AI revisited". Proceedings of the 9th conference on Computing Frontiers: 285–292.
[5] "The 7th International Computer Game Developers Conference",Computer Gaming World, 1993-07, p. 34.
[6] Hutter, Marcus (2005), Universal Artificial Intelligence, Berlin: Springer, ISBN 978-3-540-22139-5.
[7] Luger, George; Stubblefield, William (2004). Artificial Intelligence: Structures and Strategies for Complex Problem Solving (5th ed.), Benjamin/Cummings, ISBN 0-8053-4780-1.
[8] Neapolitan, Richard; Jiang, Xia (2012), Contemporary Artificial Intelligence, Chapman & Hall/CRC, ISBN 978-1-4398-4469-4.
[9] Nilsson, Nils (1998). Artificial Intelligence: A New Synthesis. Morgan Kaufmann. ISBN 978-1-55860-467-4.
[10] Russell, Stuart J.; Norvig, Peter (2003), Artificial Intelligence: A Modern Approach (2nd ed.), Upper Saddle River, New Jersey: Prentice Hall, ISBN 0-13-790395-2.
[11] The Mirage project page
[12] The Open AIR page on Mindmakers.org
[13] Constructionist Design Methodology, published in A.I. magazine
[14] Mission Engine: Multi-system integration using Python in the Tactical Language Project
[15] Poole, David; Mack-worth, Alan; Goebel, Randy (1998). Computational Intelligence: A Logical Approach. New York: Oxford University Press. ISBN 0-19-510270-3.
[16] Dowe, D. L.; Hajek, A. R. (1997), "A computational extension to the Turing Test". Proceedings of the 4th Conference of the Australasian Cognitive Science Society.
[17] Dreyfus, Hubert (1972). What Computers Can't Do. New York: MIT Press. ISBN 0-06-011082-1.
[18] Dreyfus, Hubert; Dreyfus, Stuart (1986). Mind over Machine: The Power of Human Intuition and Expertise in the Era of the Computer. Oxford, UK: Blackwell. ISBN 0-02-908060-6.
[19] Dreyfus, Hubert (1992). What Computers Still Can't Do, New York: MIT Press. ISBN 0-262-54067-3.
[20] Dyson, George (1998). “Darwin among the Machines”. Allan Lane Science. ISBN 0-7382-0030-1.
[21] Edelman, Gerald (23 November 2007). "Gerald Edelman – Neural Darwinism and Brain-based Devices". Talking Robots.
[22] Edelson, Edward (1991). The Nervous System, New York: Chelsea House. ISBN 978-0-7910-0464-7.
[23] Fearn, Nicholas (2007). The Latest Answers to the Oldest Questions: A Philosophical Adventure with the World's Greatest Thinkers, New York: Grove Press. ISBN 0-8021-1839-9.
[24] Gladwell, Malcolm (2005). Blink. New York: Little, Brown and Co. ISBN 0-316-17232-4.