iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 2 **Issue:** XII **Month of publication:** December 2014

DOI:

www.ijraset.com

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

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

Intelligent Agent Business Development Systems - Trends and Approach

P.S.Velumani
Research Scholar,Dr.M.G.R.Educational and Research Institute University,Chennai,Tamilnadu,India.
Assistant Professor, Model Institute Of Engineering and Technology,Kot Bhalwal, Jammu.

Abstract:-This paper focus on what and why of this Intelligent Systems Agents in Artificial Intelligence, How can we conceive, design, and build adaptive, enterprise-scale information systems, its business and technical benefits and implications and then how to apply the paradigm to business, explain the relationship with object technology in particular the Object Management Group's Object Management Architecture and discusses certain concerns and issues surrounding intelligent systems.

I. INTRODUCTION

Intelligent Systems can generally be defined as software programs, which assist their user and act on his/her behalf: a computer program which helps in newsgathering, acts autonomously and on its own initiative, has intelligence and can learn, improving its performance in executing its tasks.
What is to be considered an Intelligent Agent, and what is not? the following features are necessary to define a true Intelligent Agent:

A. Personalizability

An Intelligent Agent is able to adapt to its users needs, by learning from how the user reacts to the agent's performance.

B. Pro-activity

Which refers to the ability of an agent to take initiatives by itself, autonomously (out of a specific instruction by its user) and spontaneously, often on a periodical basis, which makes the Agents a very helpful and time saving tool.

C. Adaptivity

The capacity to change and improve according to the experiences accumulated. This has to do with memory and learning: an agent learns from its user and progressively improves in performing its tasks. The most experimental *bots* even develop their "own" personalities, and make decisions based upon past experiences.

D. Cooperation

Interactivity between agent and user is fundamental, differently from the one-way working of ordinary software.

II. LITERATURE SURVEY

The origins of intelligent systems lie in research done in both robotics and artificial intelligence beginning around the 1970s.[1] The first intelligent software agent is commonly thought to have been ELIZA, a 'computer therapist' application that was able to carry on a continuous conversation with a user.[2]
Despite experiments such as ELIZA, little use was made of intelligent software programs until the creation of the World Wide Web in the early 1990s. With the need to keep track of new pages being posted on the Web, the first search engine was created for the Internet.[3] It was named the 'World Wide Web Worm', and was used to discover and count the number of servers on the Web.[4] As time went on numerous similar systems known as web 'crawlers' or 'spiders', designed to seek out newly posted web pages and add them to the indexes of large search engines, became the first widely-used intelligent systems.[5]

In the late 1990s more agents were designed for use on the web, particularly in the realm of e-commerce. "RoboShopper", one of the first 'shopping bots', agents designed to help users shop over the Web by locating items and comparing prices, was introduced in 1997.[6] In addition to these advances made by agents on the Web, Microsoft also released its first interactive 'Office Assistants', which employ agent technology, in 1997.[7]

Despite the progress made in agent technology in the last decade, we are still very much considered to be working with only the first generation of intelligent systems.[8] The next generation of agents will be able to communicate and work together to a much greater degree than today's systems; however, before this can become possible standards for inter-agent communication must be developed.[9]

III. BUSINESS ORIENTED APPROACH-FUTURE

As we know the world economy is going digital and a handful of business pioneers already provide exemplars of the tactics and operation of 21st century business competitors. The best example is of virtual bookstore, Amazon.com. This digital bookstore turned an industry up-side-down; an intelligent and ubiquitous digital business was conceived and implemented globally. Amazon.com has opened thousands of virtual bookstores. These virtual bookstores are independently operated by people and organizations and are handled digitally. Once the bookstore is established, intelligent systems (dubbed Eyes and Ears) notify the virtual bookstore operator of relevant new books, and thousands of activity and sales reports are distributed to Associates each week without a drop of ink. Meanwhile, digital sales clerks gently suggest to customers browsing a particular book that others who have bought the book also bought these other titles … a little reference selling to the self-service customer. True to the notion of the virtual corporation, Amazon.com doesn't have physical book stores or warehouses. They compete with information and knowledge-based information systems and outsource the rest of the business in the industry's value chain. In the past the computer was essentially a filing cabinet, handling the affairs of the back office, programmed under the file clerk metaphor: capture, storage, and retrieval of records. Today, however, we are not only seeing the network is the computer™, now we can observe that the network is the business! To participate in the emerging digital economy, the business tasks we must ask of computers move from the back office record-keeping to the front line operations of the business. Automating different kinds of things business processes and workflows which are human, knowledge-based phenomena; and opening core business systems for direct interaction with customers, suppliers, and sometimes with competitors (i.e. the airlines industry). The new realities of business have created new imperatives for business information systems. Today's business systems must provide enterprise (and interenterprise) reach so that islands of disparate information can be integrated into a meaningful whole. They must be able to cope with the overwhelming complexity of distributed technology and an interenterprise information base. They must be open to survive a network-centric ecosystem. Rapid applications development goes without saying, and applications must be designed to embrace constant change. Business systems must be knowledge-based (not just information-based) if they are to cope with the incompleteness and ambiguity of real business processes and workflows. And they must be adaptive to meet the needs of the moment and bring productivity to an increasingly overwhelmed business user and self-service to customers. That's what is expected of IS today, a very tall order. Both object and agent paradigms focus on addressing change and complexity. Intelligent agent technology is the next logical step in moving the object paradigm forward and overcoming some of its shortcomings in commercial IS.

IV. BUSINESS OBJECT ORIENTED-FUTURE

An agent is "one who acts for, or in the place of, another." A software agent is a software package that carries out tasks for others, autonomously without being controlled by its master once the tasks have been delegated. The "others" may be human users, business processes, workflows or applications**.**

Figure 1 shows the anatomy of an intelligent agent. An agent has both "transient" (the active workspace) and "permanent" knowledge (permanent means intrinsic in this context; it stays stable from one execution cycle to the next). It has a controller (representing its head) which can be given alternative behavioral characteristics (e.g. stimulus-response- like, actor-like or blackboard-like). An intelligent agent performs tasks which have both declarative and procedural components. It can use alternative reasoning strategies, including belief management. An agent's repertoire of tasks represents its capabilities. Each task can have its procedural "how to do".

Component represented as rules, knowledge sources (rule sets), or methods.

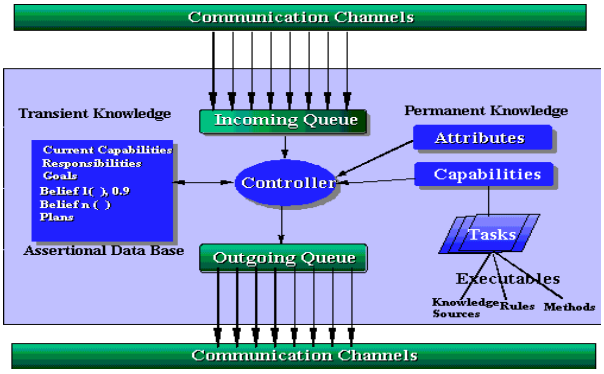


Fig. 1. Anatomy of an Intelligent Agent

A basic software agent stands on three pillars, three essential properties: autonomy, reactivity, and communication ability. The notion of autonomy means that an agent exercises exclusive control over its own actions and state. Reactivity means sensing or perceiving change in their environment and responding. And, even the most basic software systems have the ability to communicate with other entities: human users, other software systems, or objects.

An intelligent agent represents a distinct category of software which incorporates local knowledge about its own and other agents' tasks and resources, allowing it to operate autonomously or as a part of a community of co-operative problem solvers (including human users), each agent having its own roles and responsibilities.

Figure 2 shows some of the key components of multi-agent problem-solving. To achieve common goals systems need coordination. Effective coordination requires cooperation, which in turn can be achieved through communication and organization.

Fig. 2. Multi-Agent Problem-Solving

Intelligent systems are similar to objects in a number of ways. Systems can be organized into generalization and specialization hierarchies, to exploit inheritance. An agent can advertise its services using a variety of means, and how it implements these services should be transparent (ie, encapsulation of behavior). Different systems can respond to the same service request differently without the requester needing to know about such differences. For example, a user is supported by a set of E-mail systems (for Internet mail, LAN mail, etc). This user should not have to be concerned with how each agent handles the same request for sending a message (i.e., polymorphism). And, intelligent systems communicate asynchronously by message passing, using a variety of rich interaction protocols (eg, negotiation and conflict resolution protocols).

Intelligent systems differ from objects in a number of ways. Intelligent systems manipulate objects to perform their tasks. Behavior of intelligent systems (the tasks that they perform and how the tasks are performed) can be modified dynamically, due to learning or influence of other systems. Intelligent systems can be autonomous, can reason about themselves and can be mobile. Intelligent systems can actively and dynamically seek to cooperate to solve problems, using task and domain-level protocols (an important goal

is convergence on solutions despite incomplete or inconsistent knowledge or data).

Business objects make a major contribution to modeling information in the enterprise. Intelligent business systems extend this capability to provide the breakthrough in modeling knowledge in the enterprise.

The agent-oriented approach provides a more understandable paradigm for constructing models of a business domain, as well as offering techniques for creating smart business solutions-all enabled, but not driven, by object technology. The agent-oriented approach to business objects distinguishes between active, goal-seeking objects (agents) and passive objects. Agents are the centerpiece in domain modeling, together with the tasks they perform, how they interact with each other, how their actions generate events, and how they are organized around and manage business processes. This method of software construction helps reduce cognitive dissonance through the use of constructs in software that can be mapped more naturally and directly to real-world entities than basic objects. In addition, agents can be integrated frameworks that contain, in one package, specific problem-solving functions, data and control. This helps to transform the creation of applications into a high-level task of assembling pre-fabricated frameworks, as opposed to the more complex task of grappling with object libraries or fine-grained components.

Going beyond today's business objects, intelligent business systems are the next higher level of abstraction in model-based solutions to business problems. By building on the distributed object foundation, intelligent agent technology can help bridge the remaining gap between flexible design and useable applications. Intelligent systems support a natural merging of object orientation and knowledge-based technologies. Intelligent agents can facilitate the incorporation of reasoning capabilities within the application logic (e.g. encapsulation of business rules within systems or modeled organizations). They permit the inclusion of learning and self improvement capabilities at both infrastructure (adaptive routing) and application (adaptive user interfaces) levels. Intelligent user interfaces (supporting task centered user interfaces and intelligent assistance to end-users) can be a boon to productivity in a network-centric world. On the technology side, intelligent systems solve run-time operations bottlenecks inherent in network-centric computing. Management of enterprise-wide objects in a run-time environment becomes more difficult using pre-programmed control objects. For example, as the size of an OO system grows, the number of messages between objects grows non-linearly and "controller objects" themselves need to be coordinated or they can become performance bottlenecks. Unlike objects, intelligent systems can participate in high-level (task oriented) dialogues through the use of interaction protocols in conjunction with built-in organizational knowledge. In many cases, the need for communication is greatly reduced, as within these high-level dialogues, complex packets of procedural and declarative knowledge as well as state information may be exchanged in the form of mobile agents.

Figure 3 shows an example of an intelligent agent specializing in supporting risk management. It shows that such an agent would use its knowledge of risk assessment, its role in supporting this process and historical data to examine various data feeds in order to compose a risk exposure picture.

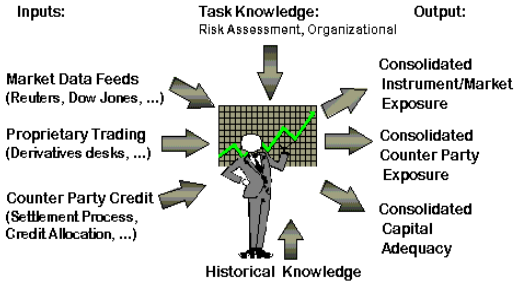


Fig. 3. Risk Management Application of Intelligent Systems

V. CONCERNS AND ISSUES SURROUNDING INTELLIGENT AGENTS

The largest single issue currently facing the further progress of intelligent agent technology is the lack of industry standards for inter-agent communication languages and agent directories. These standards are needed to establish the common ground between agents and servers, or agents and other agents, that is necessary for them to be able to communicate.[10] Currently a number of

communication standards for agents are under development. Until communication and directory standards are put into effect, intelligent agents will continue to be limited by these communication difficulties and will not be able to work together or be truly effective.

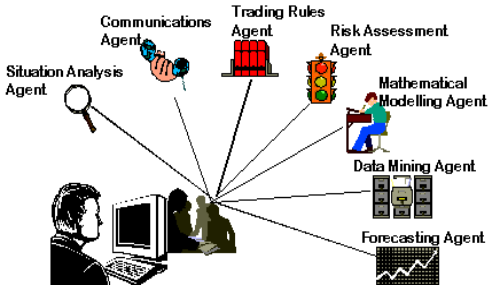


Fig.4. Intelligent Systems as Personal Assistants to a Financial Trader

A. Privacy and security

A common concern about the increasing use of intelligent systems on the World Wide Web has been the protection of the privacy and security of individuals and corporations alike. At the moment effective safeguards still need to be developed that will protect the data security of companies and the personal privacy of individuals from 'visiting' systems requesting information of a server, website or computer. An agent is "one who acts for, or in the place of, another." A software agent is a software package that carries out tasks for others, autonomously without being controlled by its master once the tasks have been delegated. The "others" may be human users, business processes, workflows or applications Building agent-based systems can be characterized in terms of three distinct, but related, topics:

1) *A full-lifecycle solution development process that explicitly addresses domain modeling* -- the effort spent in domain and problem understanding can be leveraged to guide and accelerate application development.
2) *Ontology-based domain models* -- fully understanding the domain is essential to understanding the problems in the domain. The richness of domain models determines reuse and flexibility.
3) *Architecture* -- applications must follow a consistent architecture approach so they can interoperate with each other.

VI. A FULL-LIFECYCLE SOLUTION DEVELOPMENT PROCESS

The agent-oriented lifecycle model should be iterative and incremental. Over a decade of experience taught us that agent orientation will enrich contemporary OOA/D by providing an *active* modeling paradigm and better analysis models that enable reuse. As shown in Figure 1, our agent-oriented domain modeling and architectural frameworks can be integrated with best practice software engineering lifecycles, leveraging the contributions of artificial intelligence (AI) and object-oriented software engineering. We thus preserve considerable intellectual capital while taking the next step in development methods for truly intelligent and adaptive information systems.

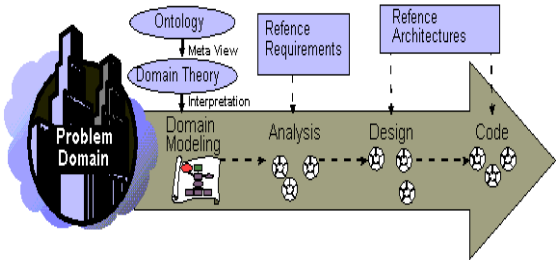


Fig.5. Agent-oriented Lifecycle

VII. ONTOLOGY-BASED DOMAIN MODELS

Building agent-based solutions emphasizes the importance of domain modeling as a critical initial step in producing business applications. There are at least three distinct approaches to domain modeling: business process reengineering (BPR), object-oriented (OO), and artificial intelligence (AI), the underpinning of intelligent agent technology (IA). All three approaches are model-based and offer techniques for describing problem domains. As shown in Figure 6, the three approaches offer differing perspectives of the problem domain. The convergence of OO, BPR, and AI technology results in a significant breakthrough in building models of the enterprise that are capable of end-to-end integration of business analysis and software systems. Although the fusion of these disciplines is not yet complete to the point of standardization, pioneering companies have recognized the competitive advantage and launched major agent-oriented development initiatives.

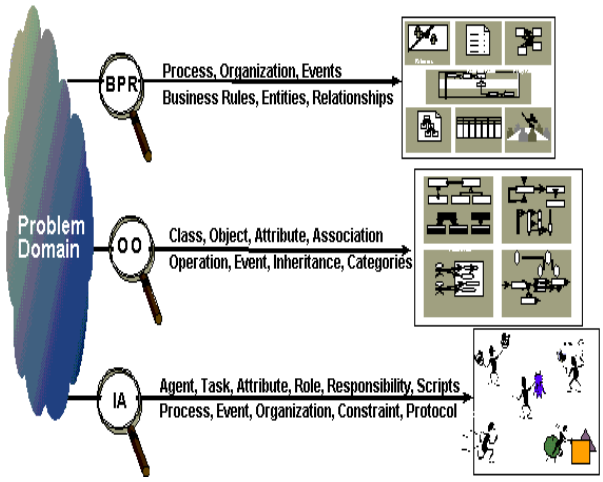


Fig.6. OO, BPR, and IA Perspectives on Business

VIII. DOMAIN MODELING

Traditional BPR methods are expressive and easily (often, naively) understood for describing business process and organization. But traditional BPR produces separate models of data, processes and organization that soon become unmanageable. The models are difficult to integrate, validate or leverage for the development of software systems. Information is captured using informal techniques, and incomplete lifecycle models lack a clear implementation path.

Object-oriented methods are well suited to software engineering and have potential for reuse. They are, however, not inherently business oriented, not sufficiently expressive of most problem domains, and often foster premature commitments to design and implementation strategies. Consequently, they do not capture a sufficiently rich domain model for significant reuse. In practice, most OO methods are informal and not repeatable. Traceability is questionable and robust design heuristics are hard to find.

Business models must make a distinction between passive and ***active, goal-seeking*** objects (intelligent agents) if we are to close the gap between model and reality. Business object models need to contain behavioral declarations, role definitions, ontological awareness and constraint axioms.

Intelligent agent technology can be leveraged to enhance enterprise modeling as well as offering new techniques for developing intelligent applications and smart technical infrastructure services. An agent-oriented perspective allows us to develop rich and expressive models of the enterprise the foundation for adaptive, reusable business software. An agent-oriented approach such as the Knowledge Analysis and Documentation System (KADS) supports richer representation of the problem domain and tends to be more rigorous and mature compared to BPR and OO. A method like KADS entails little commitment to design in early modeling phases and allows alternative implementation paths. On the down side, methods like KADS are not widely known outside the artificial intelligence community. They tend to be weaker on infrastructure issues and lack the diversity of commercial tools found in the object technology world. Traditional artificial intelligence is knowledge-base centric and can result in monolithic design.

Blending the strengths and sifting out the weakness of each discipline, Agent-Oriented Business Engineering takes us one great step forward. AOBE is a full life-cycle proposition, not just a programming or implementation issue. The secret to AOBE's success is to apply agent technology up front as a *domain modeling metaphor* while using the maturity of OOSE to provide the

infrastructure lacking in the artificial intelligence world. Figure 7 shows that agent-based domain modeling produces the business object model in terms of ontologies, and uses ideas and techniques from traditional business analysis (Process Modeling), OO (UML), and AI (Agents and Rules). While devoid of implementation details, the business object model is agent-based and is used to support user task analysis, requirements modeling, and the specification and design of the software object model.

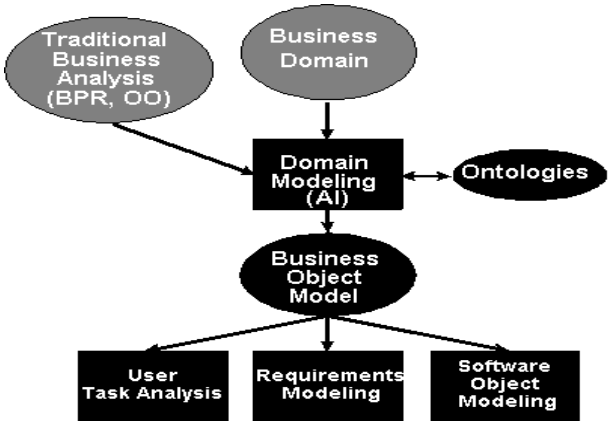


Fig. 7. Combined BPR, OO, IA Domain Modeling Process

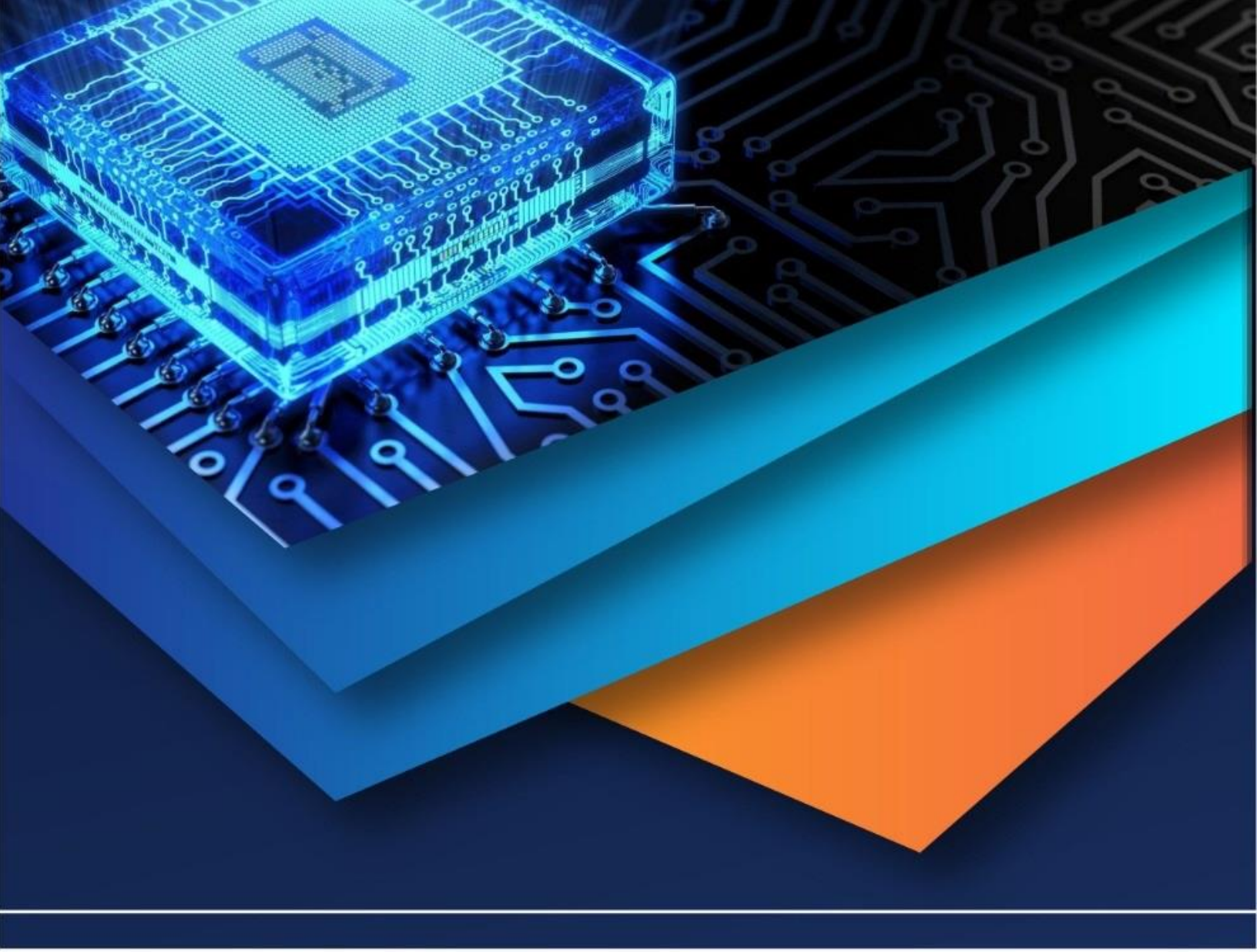
A domain model is commonly defined as a representation of entities, their attributes, their behaviors, and the processes that bind them together within a domain - nothing new to the experienced OO modeler. What is new and highly beneficial is to define the information in a domain model in terms of ontologies and then to use these to construct business object models.An ontology can be expressed in a variety of waysfor example: informally (Natural Language, Graphical Notations), semi-formally (for example, Ontolingua), formally (first order logic languages such as Knowledge Interchange Format [KIF]). The *de facto* language for encoding ontologies is Stanford University's *Ontolingua,* a portable language for writing ontologies. Ontolingua is a LISP-like language that is based on KIF. Because KIF provides for the representation of knowledge about the representation of knowledge, storing business knowledge in KIF opens up many opportunities for delivering reuse and validation at the knowledge level.

IX. CONCLUSION

The importance of Intelligent agent technology is now widely recognized in several industries, especially defense, telecommunications, and manufacturing, as well as standards bodies such as the Object Management Group. But there are certain issues like the lack of industry standards for inter-agent communication languages and agent directories, the protection of the privacy and security of individuals and corporations and ethical issues. Implementation of agent based software systems identified a number of technical issues. A mix and match software strategy has to be adopted, resulting in the need for well defined systems architecture to maintain coherence.An intelligent greenhouse can be architected to grow the software of the future. Future generations of intelligent systems will be even more complex, perhaps to the point of being able to self-replicate and create sub-systems for themselves which they can then command.[10]

REFERNCES

[1] Faltings, "Intelligent Agents: Software Technology for the New Millenium." .
[2] Ibid.
[3] Murch and Johnson, Intelligent Software Agents, 40.
[4] Martijn Koster, Robots in the Web: threat or treat?, 4 October 2001.
[5] Murch and Johnson, Intelligent Software Agents, 14.
[6] Ibid.
[7] Mack Male, Why developers should embrace Microsoft Agent - VB Square, 1997.
[8] Murch and Johnson, Intelligent Software Agents,
[9] Faltings, “Intelligent Agents: Software for the New Millennium.”40.Ibid., 66.
[10] Murch and Johnson, Intelligent Software Agents, 39.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY