
Study of an Agent Assisted New Generation Methodology

Dr. Devesh Katiar,

Assistant Professor, DSMNR University, Lucknow.

Abstract:

The objective of this paper was to produce a ready to use, complete, high expressive and a full life cycle methodology for developing agent systems. The result of this paper is the construction of a methodology for creating Software System based on multiple software agents. The imperfection exhibited by the various agent oriented methodologies led to the development of new methodologies. To name some of the drawbacks of the existing agent methodologies is that they never established fixed standards nor they were commonly expected. Their implementation phase was never explained how to implement them. Some of them never provided support for essential software. Some of them lacked complete support of agent concepts. The main concept of this paper is to build a new methodology for the development of agent system. The methodology expected to be a reliable guide for building and developing of a new system.

Agent Oriented Software Engineering

The agent oriented approach promise the ability to construct flexible system with complex and sophisticated behaviour by combining highly modular components. The intelligence of these components-the agents-and their capacity for social interaction results in a multi-agent system (MAS) with capabilities beyond those of a simple 'sum' of the agents. The availability of agent-oriented development toolkits has allowed the technology to be assessed for industrial use. Many case studies have been carried out, yielding promising results that have aroused industrial interest in the technology.

Most recent software engineering methodologies are designed for an object-oriented approach. Engineering of commercial MAS requires the availability of agent oriented software engineering (AOSE) methodologies. Most MAS systems will be implemented with object and component based technology in the near future unless a widely accepted agent programming language emerges. In this case, viewed at a detailed level, an agent is a relatively complex object or component. However, this is like considering that a house is a pile of bricks, but it is more convenient to view a kitchen and bedroom. When an agent is viewed at a more abstract level, structures come into focus, that are not found in conventional objects or components. Agent-orientation is thus a paradigm for *analysis. Design and system organization*. An agent-oriented modeling language must provide primitives for describing these higher-level structures, the inspiration for which derives from cognitive psychology and social modeling via artificial intelligence.

Methodology for Engineering Systems of Software Agents builds upon current software engineering best practices covering analysis and design of MAS which is appropriate for use in mainstream software engineering departments. It has well defines concepts and a notation that is based on UML whenever appropriate.

Agent Based System

Agents are considered one of the most important paradigms that on the one hand may improve on current methods for conceptualizing, designing and implementing software systems and on the other hand may be the solution to the legacy software integration problem. An agent is simply another kind of software abstraction. Compared to object oriented abstraction that describes methods and attributes of a software component, and agent is however, is an extremely high-level software abstraction. It provides a convenient and powerful way to describe a complex software entity. Rather than being defined in terms of methods and attributes, an agent is defined in terms of its behavior.

The agent technology can overcome some limitations of the well known client-server model regarding to the scalability as well as network delay performances issue. In agent-based concepts, the user agents are dynamically created not depending on the number of limited threshold as in the client-server concepts. Agents can be created indefinitely from the agent repository in order entertain all incoming requests.

Recently, the architecture of agent based has been widely adopted in many kind of system. The agents in the system are used to check essential requirements that clients have to meet and prepare reports for assessors. The system proves useful and helps in reducing assessment time

Existing Methodologies

Since agent-based systems became more and more used not only in the academia but also in practical industrial environments, a methodology for developing such systems became more important. Different methodologies were developed and proposed, all of them being inspired by older software engineering methodologies. There are two main starting points:

-) The Object oriented approach, having UML and U²ML as support language.
-) Knowledge Engineering approach.

The approach adopted in our work under development uses as starting point the Object-Oriented methodology and UML and U²ML.

Our approach does not consider the multi-agent system as a set of independent agents that act individually in the same environment in order to fulfill some goals. We consider multi-agent systems a set of interrelated agents that communicate to each other trying to fulfill not only their individual goals but also common goals. The agents should also respect some social laws – forming therefore a society or an organization.

The methodology contains a requirements specification phase, an analysis implementation and deployment phases are not discussed below:

-) Specification of the Functionality of the system as an organization. The functionality of the system is expressed in terms of provided services to the user. The provided services can be mapped on the social tasks of the agent organization members.
-) Designing the organizational model in terms of the role model and social rules. This means the identification of positions\functions that must be fulfilled by agents besides their individual tasks and also the social laws that should be obeyed by all the agents members of the organization.
-) Identifying the interaction patterns that represent communication protocols between different roles, therefore defining the interaction model.
-) Designing the environment model in terms of existing resources and access protocols to them.

In the design phase the concepts defined above are refined on a lower abstractization level.

The main steps are presented below:

-) Designing the agent model that should support the role model and interaction model defined above.
-) Designing a coordination model that contains the coordinated elements, a coordination media and also a set of coordination laws. This model is derived from the interaction model and the set of social rules identified above.
-) Designing the services provided by the agents. The services model should consider both the individual agent tasks and also the social tasks of the entire organization.

Main Components Identification:

The main components involved in the evaluation of departmental working of the MOEF, which have been identified as User agencies(UA), Evaluation , Type of Evaluations, and assessment engine with various parameters for the assigning of the Projects submitted by the user agencies. The relationship between the

components is formed when the project is submitted by the user agencies for evaluation and permission for the completion of the said project. The assessment of the project is of different type and contains set of selected proforma. A assessment can be created by selecting the prescribed proforma.

The assessment engine acquires the knowledge for evaluation of the projects submitted by the user agencies. The user agencies submit their own proforma according to the project they are going to start. The only thing that has to be kept in mind is the specific proforma need for the clearance of the project at state level. Pre setting can also be done web based user agencies interface. As soon as the user agencies submit the project it is ready for evaluation on the different parameters set by the department and can view the status of the project.

Main Functional Task

The next step is the analysis i.e, identifying the functional requirements of the system. We consider the main external entity that interacts with the system as the user agencies and the assessment engines. The main functionalities provided for the user agencies are to undergo the assessment of the project according to the prescribed parameters and view the end result. The assessment entity requires functionalities such as creation / scheduling the assessments, modifying the inputs creating parameters of inputs and filling down the assessment as required in the case of the project of various categories and at the end of the processing of all the assessment viewing the status of the project. These are the required functionalities to be integrated with the existing systems.

Agent Role

The system architecture is developed by transforming the functionality requirements into a set of task of the system has to perform. Each task is then assigned to one or more agents. Our design requires more than one agent such as Coordinator Agent, Personal Agent, Assessment Agent.

Coordinator Agent

Services related to this agent is creation and task distribution are under the responsibility of the coordinator agent. This agent is created on the main part of the MOEF system and has some important parallel tasks to perform. All the tasks will be implemented on separate agents of MOEF system.

The primary task of the agent is to collect all related information about the assessment of the project authorities. This agent is programmed with the inputs that enable it to receive information from the personal agents. The communication is done viva a common interface.

The secondary task of this agent is to trigger events. This task will be scheduled on regular basis where it would keep on checking for updates for the user agencies. At a scheduled time assessment agent will be called for their specific purposes. Coordinator agents will issue the instructions for automated marking to the assessment agents using common interface. Coordinator agent will also act as a interface between the user agencies personal agents that are on line and will respond with the migration of the assessment agents, which on being updated it would start automated assessing process.

Personal Agent

We have certain distinction between assessing personal agents and user agencies personal agents because they both require Personal agents to perform the individual tasks. Both type of personal agents are on the client machine and all communicate viva a common medium. Personal agents are given the responsibilities to monitor the various assessment parameters. These agents are responsible for the assessing the projects submitted by the user agency. When ever a new project is found the notification is also sent to the coordinator agent. Once the user agency logs into the system for the processing of the project, the Personal Agents of the user agencies will maintain proper communication with the coordinator agents. Once automated assessing is complete the personal agents will notify the user agency about the current status of the project.

Assessment Agents

Assessment agents will be generated from agent repository supervised under coordinator agent. Assessment agents are loaded with the automated analyzing and accessing modules. They migrate from the server to the

client machine. Assessment agent is therefore responsible for traveling to the site where the user agencies will be interacting with cooperating with the personal agent responsible for the accessing of the project. The fig 1 illustrates the overall process of the automated evaluation module of the projects in the Assessment Agents. The verification process can be done by performing similar verifications which is depicted in fig 2. The tested projects will be accepted if they pass the similar tests. Acceptance or rejection is determined by comparing the calculated similarity value with the set threshold.

The project is accessed and its measures are collected through the testing process.

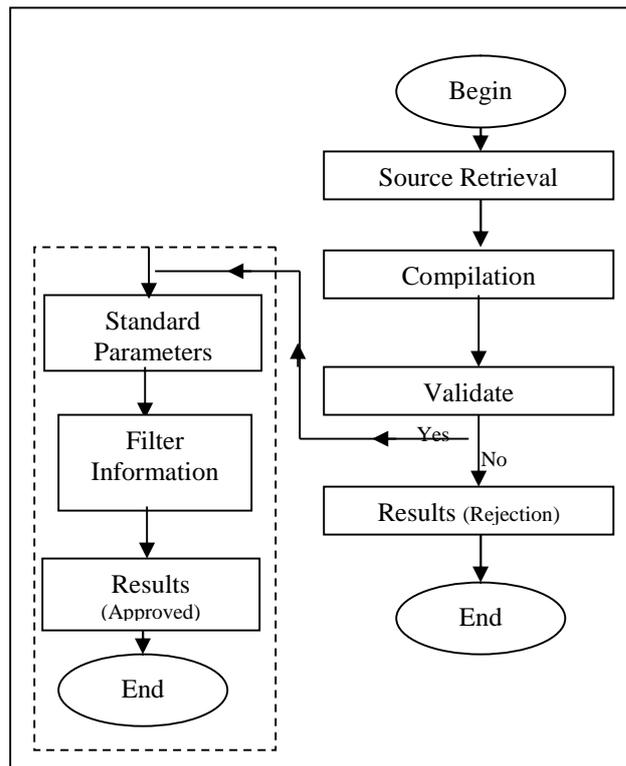


Fig 1: Automated Process of System Flow

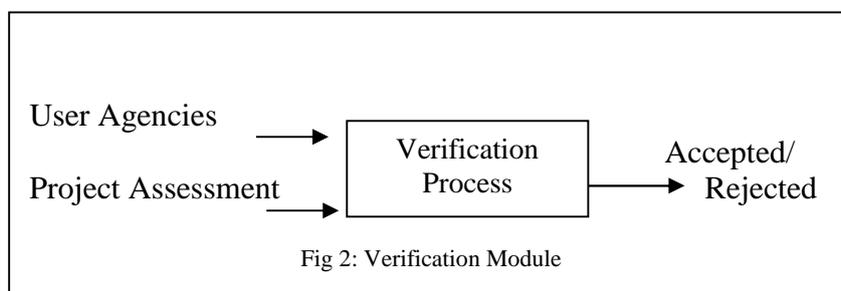


Fig 2: Verification Module

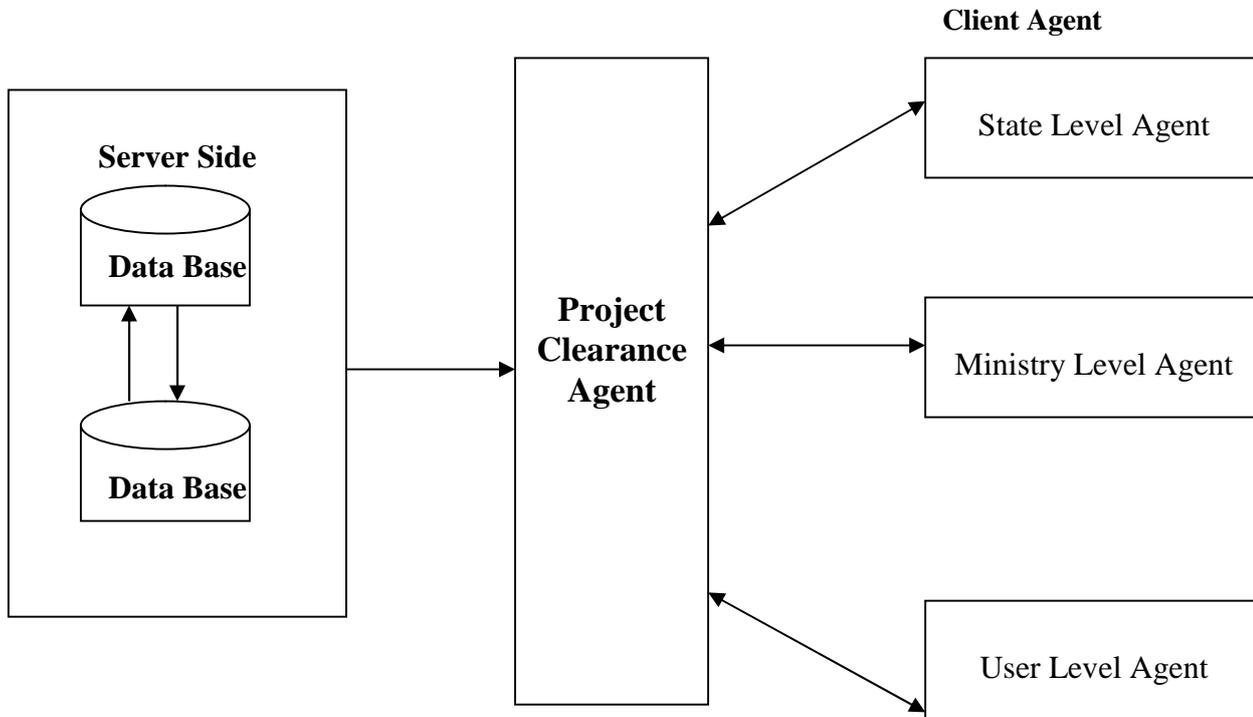
Proposed Methodology

My proposed system is based on new generation agent assisted tool. It is divided into three section :

- ❖ *State Level Evaluation*

- ❖ *Ministry Level Evaluation*
- ❖ *User Agency*

Projects are supposed to check for e-monitoring of projects from time to time . Main Architecture for this system is.



Project Assessment Methods

This process can be divided into various process which are as under.

I) *Project Evaluation at State Level.*

There are various parameters which have to be kept in mind before the project is given clearance for work. And they are evaluated that the project is not covering any Forest area. If there is felling of trees than there is alternate arrangement for the planting more than the number of trees felled. And all the alternative options have been explored or not. The total vegetation density of the said area. Clearance from the State Forest Department, Nodal Officer, PCCF, Local Bodies has been received or not. These are some of the most important parameters for the user agencies to receive before they thing of butting their project for clearance to ministry.

II) *Project Evaluation at Ministry level and According Clearance.*

After the project is submitted to the Ministry it is evaluated on the above said parameter and if any more information is required then those rectification is called by the user agencies. Before further processing of the project. If all the parameter meet the requirement it is accorded Principal Approval with few more specification and after the completion of the said conditions it is accorded final Approval.

Here the evaluation criteria are set on the different assessment agents. Functionality is measured simply by conducting the black box testing of the programs.

III) *User Agencies*

In addition to the above evaluation the user agencies are evaluated according to the project they have applied for it may i.e mining, thermal, Hydel, Road Defence and many more. A program is said to successful only if compiles all the data successfully and generates the desired results keeping in mind the accessing parameters.

Overall System Architecture

The overall system is divided into three sections: the server site, project site and user agency. In the project site it is decided that what type of project is the user agency applying for. The server site has all the detailed parameters to evaluate the project; it is done through the different agents used. The coordinator agent plays an important role between the user agency and the server site till the project is processed and we get the processed result or output.

Conclusion And Future Scope

We have designed a lightweight agent-based architecture which automated the working of the department in processing the projects submitted by the project authorities for according clearance on the respective projects.

Future work can be directed towards improving the similarity checking functions, to allow more flexibility when comparing the outputs and time it takes from the point of user agencies and later implemented at all the regional levels.

References

1. T. DeMarco. The Paradox of Software Architecture and Design. In Stevens Prize Lecture. August 1999.
2. P. Donzelli and M. R. Moulding. Application Domain Modelling for the Verification and Validation of Synthetic Environments: from Requirements Engineering to Conceptual Modelling.
3. P. Giorgini, A. Perini, J. Mylopoulos, F. Giunchiglia, and P. Bresciani. Agent-oriented software development: A case study. In S. Sen J.P. Müller, E. Andre and C. Frassen, editors, Proceedings of the Thirteenth International Conference on Software Engineering - Knowledge Engineering (SEKE'01),
4. F. Giunchiglia, J. Odell, and G. Weiß, editors. Agent-Oriented Software Engineering III.
5. F. Giunchiglia, A. Perini, and F. Sannicol'o. Knowledge level software engineering. In Springer Verlag, editor, In Proceedings of ATAL 2001, Seattle, December 2001.
6. Connie U. Smith, L.G.W.: 1. In: Introduction to Software Performance Engineering. Addison Wesley (2001)
7. Bosch J.: "Design and Use of Software Architecture", *ACM Press*, Harlow, England, 2000.
8. NATO Software Engineering Conference
9. B. Cohen, "The Education of the Information Systems Engineer," *Electronics & Power*, Mar. 1987, pp. 203-205.
10. B. Meyer, *Object-Oriented Software Construction*, 2nd., Prentice Hall, Upper Saddle River, N.J., 1997.
11. Bellifemine, F., G. Caire, And D. Greenwood, *Developing Multi-Agent System With Jade*. Wiley Series In Agent Technology, Ed. M. Wooldridge. 2007
12. Pornphol, P. An Agent-Based Quality Assurance Assessment System. In The 5th WSEAS International Conference On E-Activities. 2006.
13. Pattie, M., Agents That Reduce Work And Information Overload.