THE INTERNATIONAL JOURNAL OF SCIENCE & TECHNOLEDGE

An Approach to Enhance Quality of the Rad Model Using Agents

Fakeeha Fatima
Lecturer, Government College University for Women Sialkot, Pakistan
Maryam Javed
Student, Government College University for Women Sialkot, Pakistan
Fatima Amjad
Student, Government College University for Women Sialkot, Pakistan
Usman Ghanni Khan
Associate Professor, University of Engineering and Technology, Lahore, Pakistan

Abstract:
This paper acknowledges the Rapid Application Development (RAD) model as the finest model so far in the development industry. The RAD model requires minimal planning with a fast prototype generation approach with more customer satisfaction and less time consumption. But the way the world is moving with most complex and fast systems, we need more efficient and fast system-building models. Another pitfall of the RAD model that had led to our research is that it is fast but less quality-oriented; also, if developers and customers are not committed to the rapid-fire activities necessary to complete the system in a much abbreviated timeframe, RAD projects will fail. Our approach is to create an extension of the RAD model by changing its requirement-gathering procedures from manual ways to automated requirement gathering using agents, i.e., an entity that can perceive and act according to environments on its own. This technique tends to satisfy customers and by using agents try to make error prone, faster and non-dependable system which saves not only time but also increases quality and efficiency of model. In an automated system, we have used the environment of websites for agents to work on where various types of agents work collaboratively, such as template selection agents, menu bar agents, etc. By using this approach to extract requirements, the quality of the RAD model makes it even better, more efficient and less error prone. To do this, we replaced the manual techniques of requirement-gathering into automated prototyping approach using agents at various stages.

Keywords: *Artificial Intelligence, Goal Based Agents. Intelligent Agents, Joint Application Development, requirement gathering, RAD, Software process models, Software engineering*

1. Introduction

Today, computer has become a very important part of human life. It has taken charge of every field whether it is medical, economics, commerce, business or even art. It is considered to be the most time saving and problem solving device that helps us in executing complex, repeating and long procedures in less time efficiently **[1].** Software engineering provides well-defined procedures and systematic methodologies that help designers and developers in building software's with high quality [1]. The techniques of software engineering are being advanced and even complex with every coming day to make the quality of the software's even better and efficient. Quality is generally defined as "Fitness for purpose". As defined by International Organizations "Quality comprises all characteristics and significant features of a product or an activity which relate to the satisfying of given requirements". Numerous techniques are used to make sure the quality of software. A process model is a description of process expressed in a suitable process. By using appropriate process life cycle, help to decrease schedule-killing rework, increase customer satisfaction, and cut down the risk of untested requirements [2]. Types of software process models are:

- Waterfall
- Iterative
- Prototype
- Spiral
- RAD Model

Generic steps of software process models are

- Requirements phase
- Specification phase
- Design phase
- Implementation phase
- Integration phase
- Maintenance phase
- Retirement

Every software process model leads to another process model that conceals the flaws of preceding process models and provides something better. So far RAD is considered to be the most efficient and quick software producing model [10].

1.1. RAD Process Model

Rapid Application Development (RAD) was started by using the concepts of rapid prototyping approach that gather the essential features of a developed system by using prototype approach. Intentionally incomplete; to be modified, supplemented or supplanted is the most appropriate definition of a prototype [50]. Rapid Application Development (RAD) was first formalized by James Martin in early 90's whom believed to make a model that is more flexible and adaptable to the changing requirement of customer and developed quality assured systems with rapidly develop at minimum costs. So by using this approach less effort and time required on planning the system rather than on development. In this model, three people have to work together to produce efficient and effective work *i.e.* developer, end users and stake holders [3]. Phases of RAD model are [4]

- Requirement Planning
- User Design
- Construction
- Implementation

RAD uses two types of methodology [5]

- Phased Development
- Prototyping

1. Phased Development

In phase development whole system split into series of small actions that are developed successively. Analysis phase check the entire system, then customers and developers breaks the requirements into series of versions. After implementation of version 1, version 2 starts and follows the same steps and is preceded by version 3. This process is carried on unless and until all the versions are done. By using this methodology, user gets version 1 soon and quickly with the opportunity to enhance it more [5].

2. Prototyping

This methodology takes all the phases (analysis, design and implementation) into action concurrently. All these phases are repeatedly performed unless system is completely built. Users are active participators they test and provide their comments about systems which are analysed and redesigned if required during development. The process continues until users and developer agrees to finalize system [5].

1.2. Limitations of RAD Model

Some limitations are as follow

- Reduced scalability
- Reduced effort and time
- Less time but it compromises the quality [6].

These pitfalls make it sometime a complete failure and sometime leads to inefficient systems. It requires high expertise and also the cost is quite high due to toolsets and hardware required. It is often harder to estimate the progress of project due to no milestone that can lead to non-ending versions or a failure and disappointment for client. The customer is also committed to achieve the targeted prototype in given time; his commitment lacking can also cause failure.

Hence we say that, RAD is proven to be a valuable software process model but it still has risks that can be avoided if dealt with right mix of methodologies, tools and management.

The rest of article contains as follow: in Section 2 describes research objectives. In section 3 states review of literature. In Section 4 discussions about artificial intelligence and intelligent agents. In Section 5 a comparison between automated and manual requirement has been discussed, in section 6 research is concluded with proposed work.

2. Research Objective

The aim of our research is to eliminate the pitfalls of RAD model to make it even better, more efficient and less error prone. To do this, we replaced the manual techniques of requirement gathering into automated prototyping approach. For this, the merger of the field's artificial intelligence and software engineering strikes on our minds. And we have tried to make the process of software engineering more efficient and less time consuming with the use of artificial intelligence. For this purpose, we have used the intelligent agent technique with prototyping methodology of RAD model to assure the quality and reduce analysis and development time of RAD model.

The main objective in this research paper is to create an automated requirement gathering technique which could be used in RAD model to enhance its quality, customer satisfaction and to cut short the time frame. Requirements are extracted by using agents.

The following figure elaborates our approach

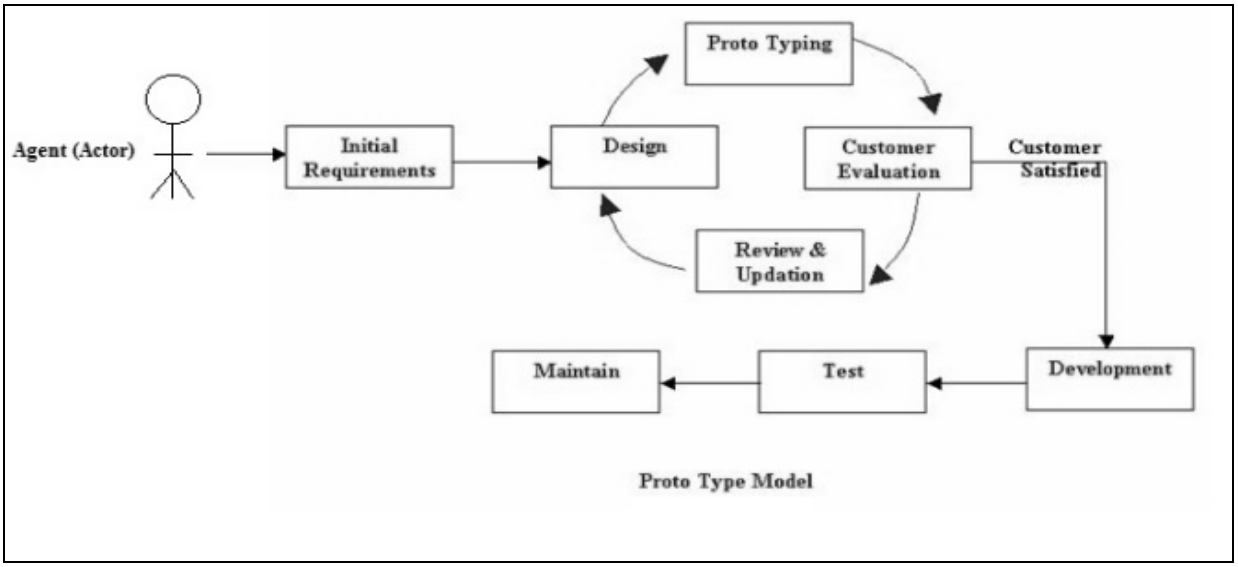

Figure: 1 Inducing agents for requirement engineering [1]

3. Related Literature

Barstow, 1987 stated in his paper that software engineering was an activity that based on knowledge of an application domain and targeted software. According to him, the software might have included many costs due to ineffective knowledge management techniques. Those costs could be eliminated by artificial intelligence techniques like heuristic search, formal inference system, rule based system and knowledge representation system. He categorized the software engineering activities into two approaches i.e. programming in large and programming in small. He stated that the application of artificial intelligence in software engineering is easily implemented in programing in small i.e. small and less complex systems whereas in programing in large, much of the work was done theoretically than practically due to lack of attention on issues like representation and use of domain knowledge, design and implementation history of software system etc. by solving these issues cost could be reduced, quality could be improved [8].

Sycara and Zeng, 1995 investigated techniques for creating the collection of intelligent agents that interact with each other to retrieve, extract and filter out all the information related to customer need and their operational environment. They described the distributed system architecture, agent's association interaction and reusability of software components. He stated three main categories of agents; i.e. interface management agents, task management agents and information dealing agents. Authors also described some coordinating mechanism for agents i.e. distributed information sources, share-ability, complexity hiding, modularity and reuse-ability, flexibility, robustness, quality of information and legacy data were also explained. Task control architecture was used for structuring of an agent; they implemented their research idea into real world as organizational decision making, health care and electronic commerce [17].

Nwana and Ndumu, 1999 provided a short perspective on achievements by research of software agent. They divided the software agent's domain into two parts i.e. multi-agent system and autonomous interface/information agent. The main purpose of paper was to evaluate the progress of agent. They highlighted the possible reasons for which agent technology was not showing desired output i.e. inability of researchers to discuss the issues associated to design, development, deployment and usage of engineering influence of system that effectively utilizes the expertise. They also explained some problems and their impact necessary to discover information, communication and ontology design their collaboration and reasoning [15].

Dipippo *et al.,* 1999 presented a real time architecture in which many agents could communicate, cooperate, negotiate and coordinate in order to achieve their goal of specific applications under particular timing conditions. Multi-agent system could benefit many applications such as military training simulations and electronic commerce etc. All these applications had specific rules of timing for operations and interaction that multi-agent might perform. They provided architecture with real time cobra layer to support real time communication and also a communication layer in which different agent could communicate using well-known agent communication languages [16].

Berger *et al.,* 2004 described in their paper that RAD model used for small to medium projects. But contrary to the research believes, some complex organizations were implementing RAD model as an approach of complex development. In this paper, they studied that using RAD model for complex systems, need of active customer interaction, team's empowerment for forming decisions, frequent distribution of products, aptness for business purposes, iteratively and incremental delivery on time, flexible to changes, higher level requirements, integrated testing analysis during lifecycle and sponsors co-operation were needed. They concluded that success of RAD model for complex systems required strong effective managerial skills [12].

Munassar and Govardhan, 2010 studied the approach concerned about the software development using different process models. They made comparison among five software process models i.e. waterfall, iteration, v-shaped, spiral and extreme programing. The five models were discussed briefly with their pros and cons. It was analysed that no process model was without problems i.e. problems of suitability and errors. Hence, analysis showed that every model had its own specialty according to varying type of customer demands, budget, and time-frame of software. Also, it exhibited that every new model introduced in the market was an extension for the older one that could remove the errors or deficiencies present in the previous one [10].

Sabale and Dani, 2012 compared process models to show more efficient model and also they stated that the making of automatic systems should be practiced to make transition between manual and automatic system easy. They used tabular way to show comparison of models like waterfall, v-shaped, RAD etc. On the basis of certain parameters like cost, risk and expertise they conclude that every model not suitable to all type of software. While choosing an appropriate model, we should study the detail conditions and requirements of system and then choose a model that could satisfy customer and system's requirements. Also, they concluded that there should be trade-off between quality and time to be consumed as customer needs a quality system in less time [11].

Harman, 2012 explored the common and closely associated work and relationship between software engineering and artificial intelligence. According to him, artificial intelligence was about to constructing intelligent machines while software engineering concerned the activity of describing, planning, designing and then applying at complex systems. He also discussed that by using artificial intelligence techniques in software engineering are very beneficial and helpful to design efficient system. Some areas where artificial intelligence (AI) was implemented are considered to be useful for probabilistic outcome, their classification, learning and forecasting in software engineering and exploration based software engineering. According to him, there is an association among machine learning approaches to software engineering and SBSE (Search Based Software Engineering). He pointed out some challenges ahead searching strategies rather than instances, exploitation of multicore computing , giving understanding to software engineers, compiling smart optimization into deployed software and innovative artificial intelligence responsive software development and deployment environment[13].

4. Artificial Intelligence and Intelligent Agent

Artificial intelligence is the study of achieving goals using intelligence that make it possible to perceive aims and act accordingly. It is not the aim of AI to build intelligent machines having understood natural intelligence, but to understand natural intelligence by building intelligent machines [18].

An agent is a system viewed as perceiving from its environment by using sensors and performing actions on that environment through effectors that maximize the probability of achieving goals [19].

Agent= Agent Program + Architecture

Agent is specified by the work it does that is called function. Function of an agent depends on what it percept[20].

F: P ∗ →A (P →Percept, A→Action)

Agent runs on the physical architecture of the system from which it perceives. An agent takes a single input and is written according to the functionality of the agent [20].

Agents are reactive, autonomous, goal oriented, temporarily continuous, communicative, and learning [21]. Various types of agents like: simple reflex agents, model based agents, goal based agents and utility based agents.

In this study we use goal based agent which are special purpose agents that accomplish their goal by following specific rules and information available. Goal based agent are allowed to select from numerous possibilities, choosing the one which reaches a goal state [22].

The system we intend to develop require multiple agents to work together to perform various tasks. Some agents may depend on others for fulfilment of their tasks. On the other hand, others might be independent on their work. A multi-agent is a loosely joined system of various software agents that can work together and cooperate with each other to perform some task or to solve some problem. These systems can work more efficiently than any individual system [23].

4.1. Requirement Gathering

Requirement gathering is the crucial phase of RAD model because the development of successful project depends on the requirements given by the user. It is most important how well you perceive your client and understands their needs. No matter, how efficient and complex system you make, if it is not following your customer's requirements then it will completely fail.

Requirement Engineering is a branch of software engineering related to real world goals, functions and constraints on software system. The relationship among these factor influences to precise specifications of software behaviour and their progression over time across software families [24].

During requirement engineering process various steps are followed that are used to engineer the requirements. It takes the customer's requirements as input, apply some operations on it and gives the requirements that later become the base of developing a system [25].

- Operations performed on requirements are:
  - Requirement Feasibility Study
  - Requirement elicitation and analysis
  - Requirement Validation
  - Requirement Management
  - Requirement Documentation [26].
- Traditional approaches for requirement gathering are:
  - Reading existing documents
  - Interviews
  - Questionnaires
  - Meetings [27]
- Collaborative Approaches for requirement gathering are
  - Group Techniques
  - Joint Application Development (JAD) Techniques

- Prototyping [27]

The latest and most efficient approach is prototyping that is a process of building a model of the proposed system. In its requirement gathering process, the analyst gathers information about the current system and processes being used by the organization. This helps the analyst understand the system and build an initial set of requirement. It presents a limited working model of the propose system If the user or organization finds it not appropriate, they reject and analyst can again work on gathering requirements or modifying the current prototype. It saves the time and effort going to be wasted after the development of complete system if user doesn't find it appropriate. It reduces the stress of technical drafts or paper work to higher tech operative systems using Computer Aided Software Engineering (CASE) [28].

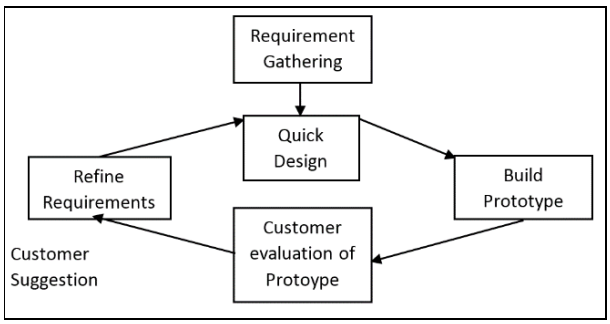

Figure 2: Prototyping Approach for Requirement Gathering [29]

4.2. Research Methodology

Our research is based on the automated requirements gathering technique to enhance the quality of RAD Model. We induce intelligent agents to gather requirement automatically that makes it more efficient, less error prone and less time taking. By combing the manual and automated approaches and made a comparison to prove that agent based approach is more efficient and less time consuming.

4.3. Automated Requirement Gathering Using Agents

We have discussed the traditional and collaborative techniques of requirement gathering. The last and most useful approach we studied was prototyping that could reduce the chances of errors, failures and customer dis-satisfaction. We are replacing the manual procedures of prototyping by the fully automated intelligent agents that speeds up the requirement gathering and analysis procedure and generates a quick design (prototype). A mini website is presented to the customer from where customer select the items by clicking on icons and agents perceive all requirements and help the user to create a prototype. If the user likes prototype, she/he can continue and fill details. Once the prototype is made, user is given options to change requirements according to his varying requirements. This procedure cuts short the time to 1/10 of actual RAD model requirement gathering procedure and improves the customer satisfaction and involvement.

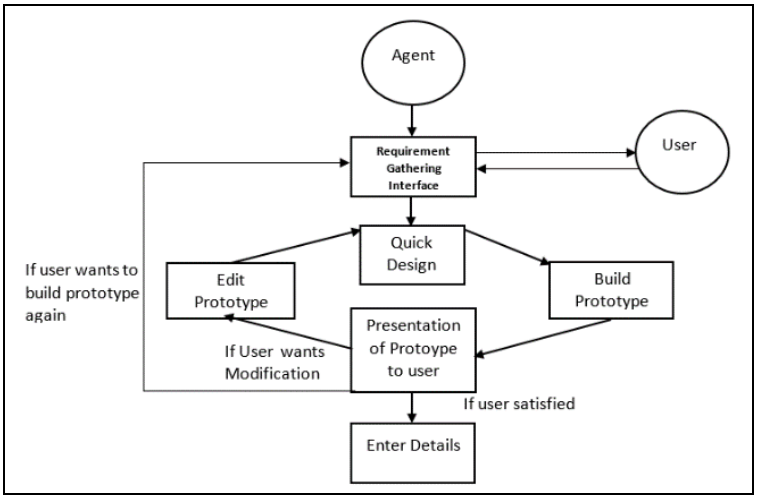

Figure 3: Automatic Prototyping through agents [28]

5. Case Study

Let's take a case study to explain how the agents enhance quality of RAD model? We'll study a case of a customer who wants to build a website for their newly built school. The customer is an end user with no deep knowledge of web technicalities. How will the designer take requirements from the customer? And also a thing to be taken in notice is that the customer wants his project to be completely done and delivered within a month. The customer starts his work by sending the questionnaire to the customer.

First of all, the question is which model to use for this kind of project because designer have RAD model with minimum time period of 3 months. Designer can't take RAD because it does not let him deliver the project in 1 month. If it takes RAD model and

minimizes the stages time then quality will be minimized. Now the developer is looking for a better option that the time of the project can be minimized without compromising the quality.

In Requirement Engineering phase, customer is asked manually all the requirements, and then a long procedure of extracting requirements occur. To avoid this over helming burden, a mini webpage is design to get requirements from customer that will controlled by multiple agent, he can send this online.

Customer receives the mini website, he login and then he's given a series of question with all built example and templates of websites. He selects his preferences and sends the requirements to the designer by choosing required templates and design. Note: Mini website can be used multiple times and upgraded according to needs.

Now, agents get the requirements and perceive the actual requirements of the customer and make a report that is sent to designer so that, he can move forward with design. It will help both developer and end users to understand actual needs. All this hardly takes 1 to 2 days that is 1/3 of the effort when requirements are taken manually. And later agents can also be used in all phases like planning, analysis, designing etc.

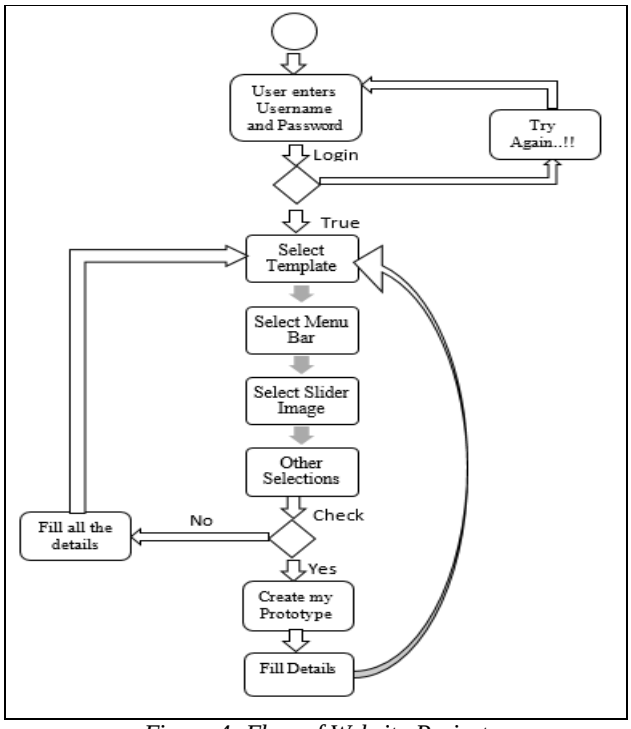

Figure 4: Flow of Website Project

5.1. Role of agents in Web Designing using RAD Process Model

The role of software agents is very significant in web designing that perform their tasks independently on behalf of developers. It can achieve its goal without any outsource interference. Agent role is defined as instances that have attributes, objectives, strategies, actions, permissions and protocols [29].

We are going to use 5 main agents for different goals. These agents are interlinked with each other as they made up a multi agent system.

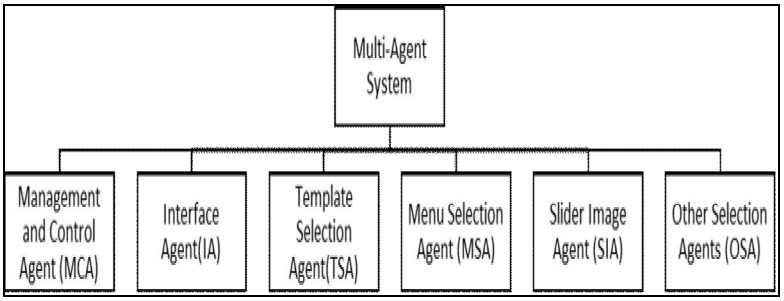

Figure 5: Selection Agents

5.1.1. Management and Control Agent (MCA)

Working of Management and Control Agent (MCA) is very important in this automated requirement gathering method. It manages and controls all the working of system. Main working of Management and Control Agent (MCA) is to receive selected information from different agents and then save them all into the database so that user can get the required output. It also debugs and monitors all the activities. Roles of MCA are:

- Attributes: Manage and control all the activities and workings of agents.
- Goals: Receives, debug and monitor all the phases and information.
- Plans: Receives the information sent from Template Selection Agent (TSA), Button Selection Agent (BSA), Menu bar Selection Agent (MSA) and Other Selection Agent (OSA) then save it into the databases.
- Actions: save the data into database.
- Permissions: debug, monitor, control and manage.

5.1.2. Interface Agents (IA)

Interface agent (IA) is one of two main agents working on the website. Interface agent is supposed to take all the data entered by user from Template Selection Agent (TSA), Button Selection Agent (BSA), Menu bar Selection Agent (MBSA) and all other selections, and then apply it on the interface in a collective form. It is dependent on other agents.

Roles of Interface Agents (IA) are:

- Attributes: Provides an interface.
- Goals: Give an interface for interaction in order to build a complete website.
- Plans: Use different templates, menus and buttons for this approach.
- Actions: On just click it transfer all selected data to MCA
- Permissions: Select, send

5.1.3. Template Selection Agent (TSA)

Template selection agent is used to record the selected template from variety of templates provided in the mini website. And then it passes the selection to the interface agent that operates it further.

Roles of Template selection agents are:

- Attributes: Provides a set of templates.
- Goals: is to provide multiple templates.
- Plans: on selection of specific template it starts working.
- Actions: Sends the selected data to Interface Agent (IA)
- Permissions: Select and sends data

5.1.4. Slider Image Selection Agent

The nature of this agent is same as template selection. It is also used to select the images which the user want to have on his website. Its work is just to take the selection to interface selection agent and then this agent is killed until the next user comes.

Roles of slider image selection agents are:

- Attributes: Provides an interface.
- Goals: is to provide multiple options to upload images
- Plans: on selection and uploading images.
- Actions: Sends the selected images to Interface agent (IA)
- Permissions: select and send

5.1.5. Menu Selection Agent(MSA)

Menu Selection Agent is used to carry the selection of menu Bar to the interface agent.

Roles of menu selection agents are

- Attributes: Provide an interface.
- Goals: Is to provide menu bars.
- Plans: On selection of menu bar it starts working.
- Actions: Sends that selected button to Interface agent (IA)
- Permissions: Select and send

5.1.6. Other Selection Agent (OSA)

Other selections may include colour scheme, font style and size, image sizes *etc.* They carry from customer to customer.

Roles of other selection agents are

- Attributes: Provide an interface.
- Goals: Provides different font styles, colour schemes for backgrounds and foregrounds
- Plans: On selection of required data it starts working.
- Actions: Sends that selected information to Interface agent (IA).
- Permissions: Select and send

5.2. Working of Multi-Agent

The agents work together and manage all activities to extract requirements. To gather requirements from customer by selecting appropriate template help designer to understand the needs and develop accordingly.

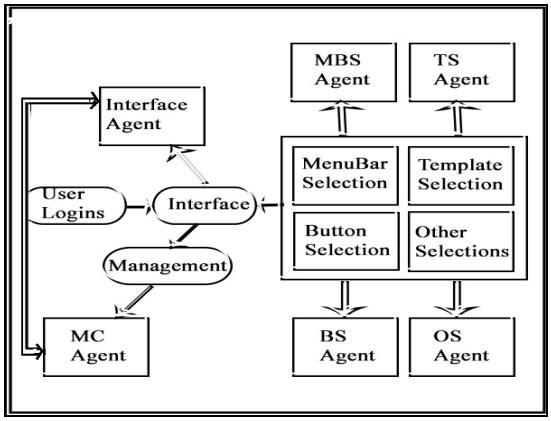

Figure 6: Working of Multi-Agent System

User enters website, login and is be provided an interface where he/she selects the template, then menu bars, buttons and other selections are done. All the selections are passed to the specific agents. Then all the information is passed on to interface agents who upload all the selections onto the webpage on specific locations. Management and Control Agent (MCA) manage all the working of whole systems.

6. Comparison Between Automated And Manual Requirement Gathering

Manual RAD model emphases on iterative and incremental delivery of the software to the customer.

The prototyping approach causes more customer satisfaction but it consumes a lot of time. We improved the model by using artificial intelligence and automated approaches which resulted in fast, error prone and less time consumption to develop software.

Automated Requirement Gathering	Manual Requirement Gathering
Less time period consumed	More time period Consumed
More efficient	Less efficient
More accurate	Less accurate
Less errors	More errors
More customer understanding	Less customer understanding
GUI Interface	Non-GUI Interface
More Customer Satisfaction	Less Customer Satisfaction
Agents are used.	Required more skilled and technical developing teams.

Table 1:Comparative analysis of automated and manual requirement gathering.

7. Conclusion

Although RAD Model is proved to be the most efficient and fast software building model so far, it still has some pitfalls like more system failure chances, less efficient in fulfilling the customer's satisfaction and more time period consumed while working on different phase *i.e.* Requirement Planning. But if we use Agents in the phases of RAD model *i.e.* Requirement planning, they can improve the quality, speed up the software engineering procedures and can produce fast , quality oriented and efficient computer systems.

8. Proposed Work

We intend to work on all the phases of RAD Model through agents. We intend to induce agents throughout all the RAD strategies and procedures (phases) to decrease the time period more and more and to provide more efficient, accurate and customer satisfactory software.

9. References

1. N.M.A. Munassar and A. Govardhan , "A comparison between five models of software engineering ", International Journal of Computer Science Issues (IJCSI), Vol.7, Issue.5, pp.94-101,2010.
2. A. Finkelstein, J. Kramer, B. Nuseibeh, "Software Process Modelling and Technology", Research Studies, 1994.
3. Hilary B., Paul B.D. and Pat C., "The utility of a rapid application development (rad) approach for a large complex information systems development", ECIS Proceedings, 2004.
4. Beynon D., Carne C., Mackay H. and Tudhope "Rapid application development RAD : an empirical review", European Journel of Information System, 1999, pp.211-223.
5. Alan D. and Barbara H.W., "Systems Analysis and Design", Ch.1,pp.8-14.
6. P. Konstantinou, "EfRapid application development" , International Society for Ecological Economics (ISEE), ISAM 5635 ,pp.1-10.
7. H. S.Nawana and D.T. Ndumu , "A perspective on software agents research",Proceeding software agents and soft computing: towards enhancing machine intelligence, concepts and applications,UK, pp.1-16, 1999.

8. D. Barstow,"Artificial intelligence and software engineering", Proceedings of the 9th international conference on Software Engineering , IEEE Computer Society Press Los Alamitos, CA, USA, pp.200-211, 1987.
9. S.T. Acuna and X. Ferre , "Software process modeling" ,In proceeding of: World Multi conference on Systemics, Cybernetics and Informatics, (ISAS-SCIs) Orlando, Florida, USA, Vol. I: Information Systems Development ,pp.1-6, 2001.
10. N.M.A.Munassar and A. Govardhan , "A comparison between five models of software engineering ", International Journal of Computer Science Issues (IJCSI), Vol.7, Issue.5 , pp.94-101, 2010.
11. R. G.Sabale and A.R. Dani , "Comparative study of prototype model for software engineering with system development life cycle ", International Journal of Engineering (IOSRJEN) , ISSN: 2250-3021 Vol.2, Issue.7, pp.21-24, 2012.
12. B. Hilary , B.D. Paul and C. Pat, "The utility of a rapid application development (rad) approach for a large complex information systems development", ECIS Proceedings,Turku, Finland,pp.220-227, 2004.
13. M. Harman, "The Role of Artificial Intelligence in Software Engineering" 1st International Workshop on Realizing AI Synergies in Software Engineering (RAISE),London,UK,pp.1-6 ,2012.
14. T. Xie , "The Synergy of Human and Artificial Intelligence in Software Engineering"The International Conference on Software Engineering (ICSE),Workshop on Realizing AI Synergies in Software Engineering (RAISE), USA, pp.4-6, 2013.
15. H. S.Nawana and D.T. Ndumu , "A perspective on software agents research",Proceeding software agents and soft computing: towards enhancing machine intelligence, concepts and applications,UK, pp.1-16, 1999.
16. L.C. DiPippo, E. Hodys and B. Thuraisingham, "Towards a real-time agent architecture – a whitepaper", Workshop on Object-Oriented Real-Time Dependable Systems (WORDS), Proceedings. Fifth International Workshop, California,USA, pp.59-64 ,1999.
17. K.Sycara and D. Zeng ,"Coordination of Multiple Intelligent Software Agents", International journal of cooperative information systems, pp.1-31 ,1995.
18. J. Rech, K.D. Althoff , "Artificial Intelligence and Software Engineering: Status and Future Trends "Special issue on artificial intelligence and software engineering, KI, pp.1-8, 2004.
19. Russel, S.J. and Norvig, P., "Artificial Intelligence: A Modern Approach", Prentice-Hall, Englewood Cliffs, N.J. ,1995 .
20. Bernhard B., "Introduction to Artificial Intelligence :Intelligent Agents ", University Koblenz, Landau , 2004-2005 .
21. Pinar D., "Artificial Intelligence: Intelligent Agents ", Handouts for CS 461,Bilkent University, Ch.2, 2008 .
22. Hilary B., Paul B.D. and Pat C., "The utility of a rapid application development (rad) approach for a large complex information systems development", ECIS Proceedings, 2004.
23. Michael W., An Introduction to Multi-agent Systems. UK:University of Liverpool,2002.
24. Vasant H., "Principles of Artificial Intelligence: Goal-Based Agents ", Iowa: Atanasoff Hall Iowa State University Ames,pp.1-11, 2005.
25. Bashar N. and Steve E., "Requirements Engineering: A Roadmap ", Proceedings of the Conference on the Future of Software Engineering, ACM ,2000, pp.35-46.
26. Ian S., "Software engineering: requirement engineering", Addison-Wesley , Inc., 9th Ed,,pp.82-111.
27. M. Wooldridge "An Introduction to MultiAgent System"s Second Edition,by John Wiley & Sons, Published May 2009.
28. Rajeev P. N., "Software Life Cycle Models", 2008.
29. Parthenopi K., "EfRapid application development" , International Society for Ecological Economics (ISEE), ISAM 5635 ,pp.1-10.