INCREASING THE ARCHITECTURES DESIGN QUALITY FOR MAS: AN APPROACH TO MINIMIZE THE EFFECTS OF COMPLEXITY

Howayda Abdallah Ali Elmarzaki and Tawfig M. Abdelaziz

Department of Software Engineering, Benghazi University, Benghazi, Libya

ABSTRACT

The efficiency of multi agent system design mainly relies on the quality of a conceptual architecture of such systems. Hence, quality issues should be considered at an early stage in the software development process. Large systems such as multi agents systems (MAS) require many communications and interactions to fulfil their tasks, and this leads to complexity of architecture design (AD) which have crucial influence on architecture design quality. This work attempts to introduce approach to increase the architecture design quality of MAS by minimizing the effect of complexity.

KEYWORDS

Multi agent system (MAS), a general architectures, Quality attributes, Recommendations systems (RS).

1. INTRODUCTION

MAS belong to the field of Artificial Intelligence, the study addressing the approaches of construction of complex systems using a large number of entities, which alter their behavior in order to accommodate with a particular problem[1], [2]. An intelligent agent can be reactive and proactive,[3] because it responses to the actions and alteration which appears in the working environment, can tack the initiative to establish the goals and interacts with other agents[4], [1],[5]. Most literatures refer that the complexity emerges clearly in architecture design of multi agent systems that assigned many and different tasks[6], [7], [8].The research work introduces an approach to increase the architecture design quality of MAS by minimizing the effect of complexity. The solution mainly presents a set of guidelines including the influential factors on the complexity of architecture design. These factors are extracted from several sides of AD. Several factors and guidelines are presented to decrease the complexity in architectures of multi agent systems. Each FG is established based on developer's previous practice or experimental methods. The FG is extracted from concepts which related to software architecture and they are presented as symbols used in application phase. For example, depending on FGM1 the hierarchical decomposition approach can be applied on books recommendation system to demonstrate the main components in visual manner to increase the understandability. The

Natarajan Meghanathan et al. (Eds) : NLP, JSE, CST, SIP, ARIA - 2018
pp. 63– 74, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80206

modularity has a major role in decreasing the complexity in software design since the interaction among agents to accomplish their tasks can lead to system complexity. Thus, this approach increases the architecture design quality of MAS by minimizing the effect of complexity. The reduction of complexity from AD, eventually reinforces the reusability concept.

2. PROPOSED SOLUTION APPROACH

- The proposed solution is to achieve the desired goals of this research work. It mainly presents a set of guidelines including the influential factors on the complexity of architecture design. These factors are extracted from several sides of AD which should be taken into consideration at the early stages of developing the architecture.

- The sides represent concepts (Abstraction, Modularity and Modeling) which be able applying in both analyses and design phases. Figure1 illustrating the approached conceptsin FG4Complexity approach.

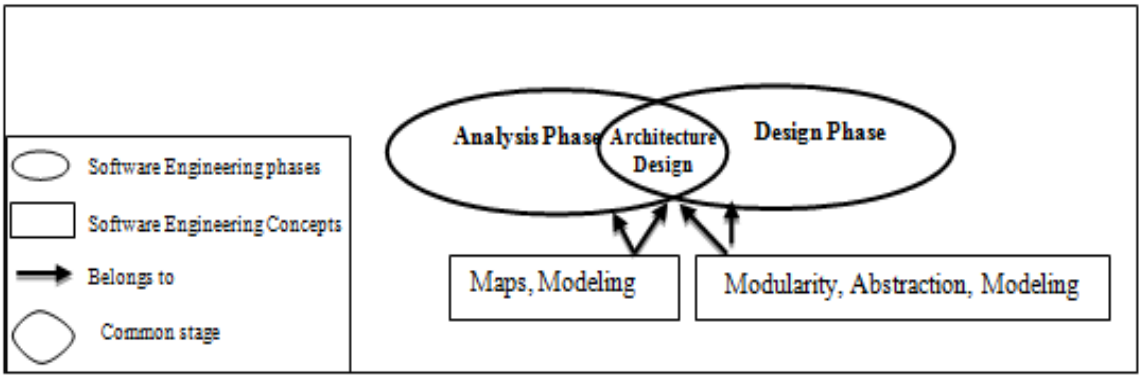


Figure 2: The concepts of analyzing and design which were addressed in FG4 Complexity approach.

- To label the proposed solution approach we suggested that "FG4Complexity". Thereby, "F" liter means Factors, "G" liter means Guidelines, and the "number 4" means for. The next figure shows the proposed approach mechanism.

- The work will be applied via some models used in methodologies related to agents systems such as HLIM[9], MASD [10].

2.1 Factors and Guidelines (FG)

In this section several factors and guidelines are presented to decrease the complexity in architectures of multi agent systems. Each FG is established based on developer's previous practice or experimental methods. The FG is extracted from concepts which related to software architecture and they are presented as symbols used in application phase. For example, the FG is related to modeling concept and represented by FGMOD symbol. The FG is related to abstraction concept and represented by FGA symbol and the FG is also related to modularity concept and represented by FGM symbol. Also, each FG should be numbered for example, FGA4 means the

factor and guideline number4 in abstraction concept section, FGMOD2 means the factor and guideline number2 in modeling concept section as illustrated in the table below.

Table1: The symbols interpretation of architecture concepts

Instances	Symbols Interpretation	symbols	Architecture Concept
FGA1….i where I is Integer number	Factors and Guideline of Abstraction	FGA	Abstraction
FGM1….i where I is Integer number	Factors and Guideline of Modularity	FGM	Modularity
FGMOD1….i where I is Integer number	Factors and Guideline of Modeling	FGMOD	Modeling

Factors and Guidelines for Abstraction (FGA)

FGA1. Developers should use Simplifying Abstraction type if they want to decrease the dynamic complexity type. [11]

FGA2. Choosing the appropriate level of abstraction.[12]

FGA3. Avoid to adopting the concept of (gold plating).[13]

Factors and Guidelines for Modularity (FGM)

FGM1. Using Hierarchical Decomposition Approach (HDA) which considers a major method of handling complexity in conventional software analysis and design. [6], [14], [15]

FGM2. It is useful to establish the software modularity based on roles or measurements such as Cohesion Communication Measurment (CCM). [16]

Factors and Guidelines for Modeling (FGMOD)

FGMOD1. Using Use Case Maps (UCM) to clarify the most relevant, interesting, and critical tasks of MAS system. [17]

FGMOD2. Using simple notations is very important to enhance understandability and decrease complexities in AD such as arrows, components, domains…etc. [18]

3. CASE STUDY APPLICATION STEPS AND DISPLAY THE RESULTS

The case study is a "books recommendations system" based on MAS to help users select books. The system can switch to three recommendation approaches Content-based filtering approach (CBF) [19], [20] Collaborative Filtering approach (CF) [21], [22] and knowledge based approach (KBA).[23], [24] The agents within the system can exchange the messages among each other via one of agent communication languages. In this case study, the messages exchanged will be via Knowledge Query and Manipulation Language (KQML).The work will be applied via some models used in methodologies related to agents systems such as HLIM[9], MASD [10].

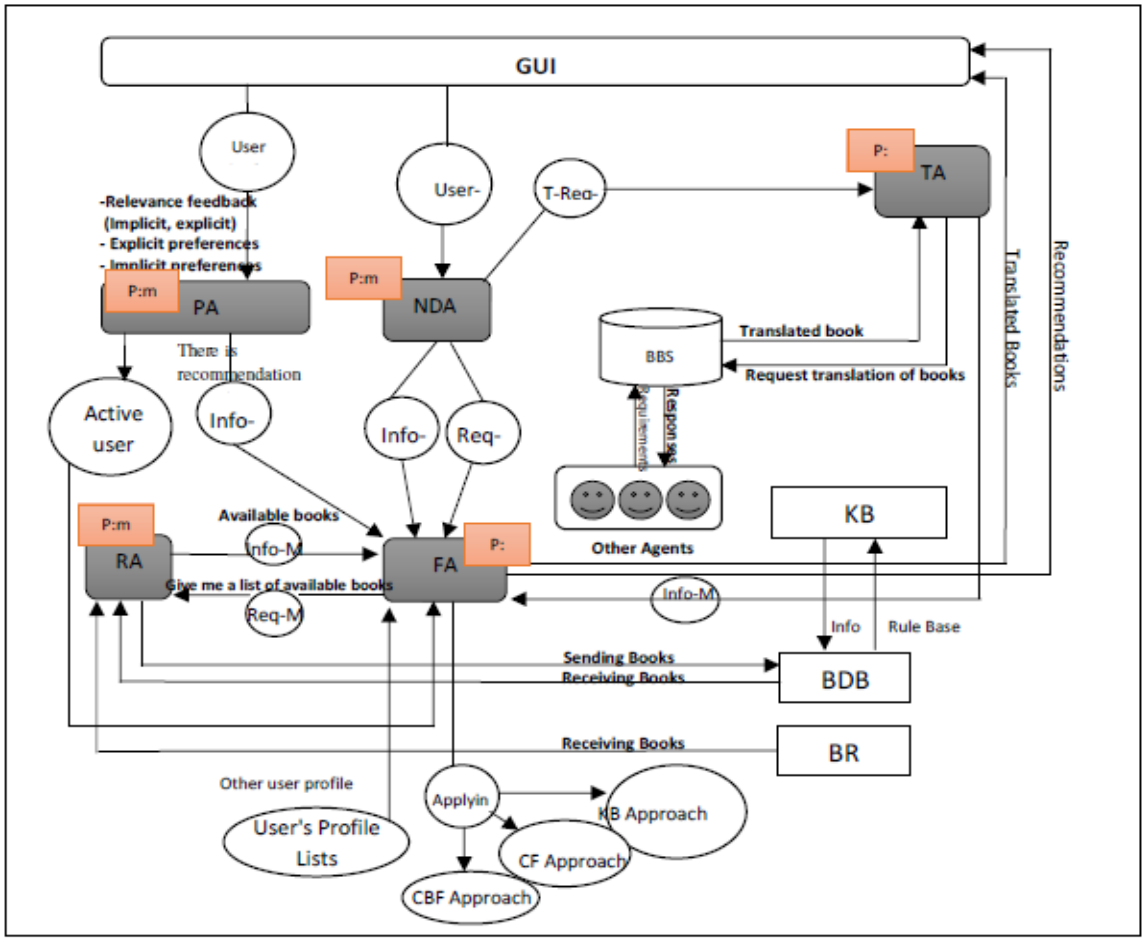
3.1 Agents and Their Tasks

A brief summary of agents and their tasks in the next table:

Table 2: The agents and their tasks

Agents	Roles (Tasks)
Profiling agent	• Gathering the user's preferences, gathering the relevance feedback, and building and updating the active user profile
NDA	Gathering the user current needs
Filtering agent	• Producing the recommendations, removing the books that are not currently offered from the recommendation list, and transferring the recommendation to the GUI
Retrieval agent	• Retrieving the books that are currently offered from the books database and storing the available books in the recommender system database
Translation agent	• Producing books translation service for users

3.2 Conceptual Overview of Books Recommendations System Architecture Design



3.3 The FG4 Complexity Approach Application Strategy

As we have earlier pointed out that all the previous FG will be within 4 steps to correspond to the current case study as the next figure shows:

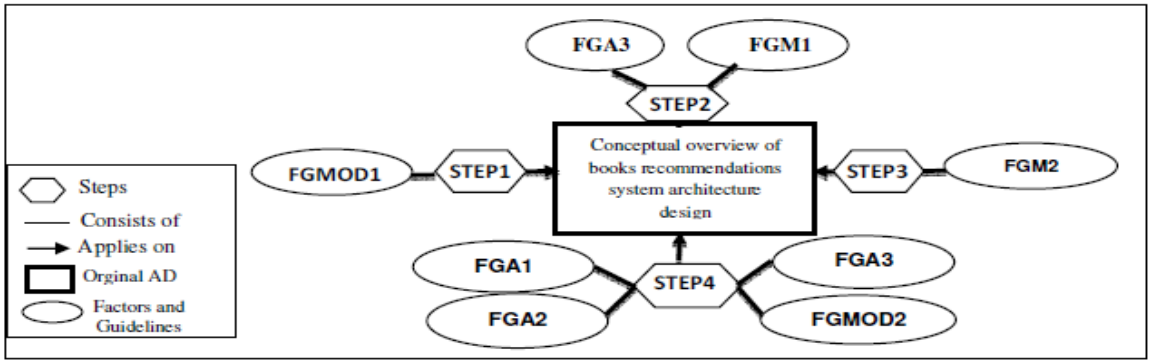


Figure 3: Illustrating of the applied steps on AD

Step1. Initially, this step is based on applying UCM represented in FGMOD1 of FG4Complexity approach which used in between analysis and design phases. These maps give high view of system specifically the responsibilities (Tasks) and interactions in a simple way, reinforce system understanding and overcome some situations of complexity such as intercommunication among agents. The following figure illustrate example to use the use case maps in analysing agents, tasks, scenarios and the most significant interactions among agents in books recommendations system. [25], [26]

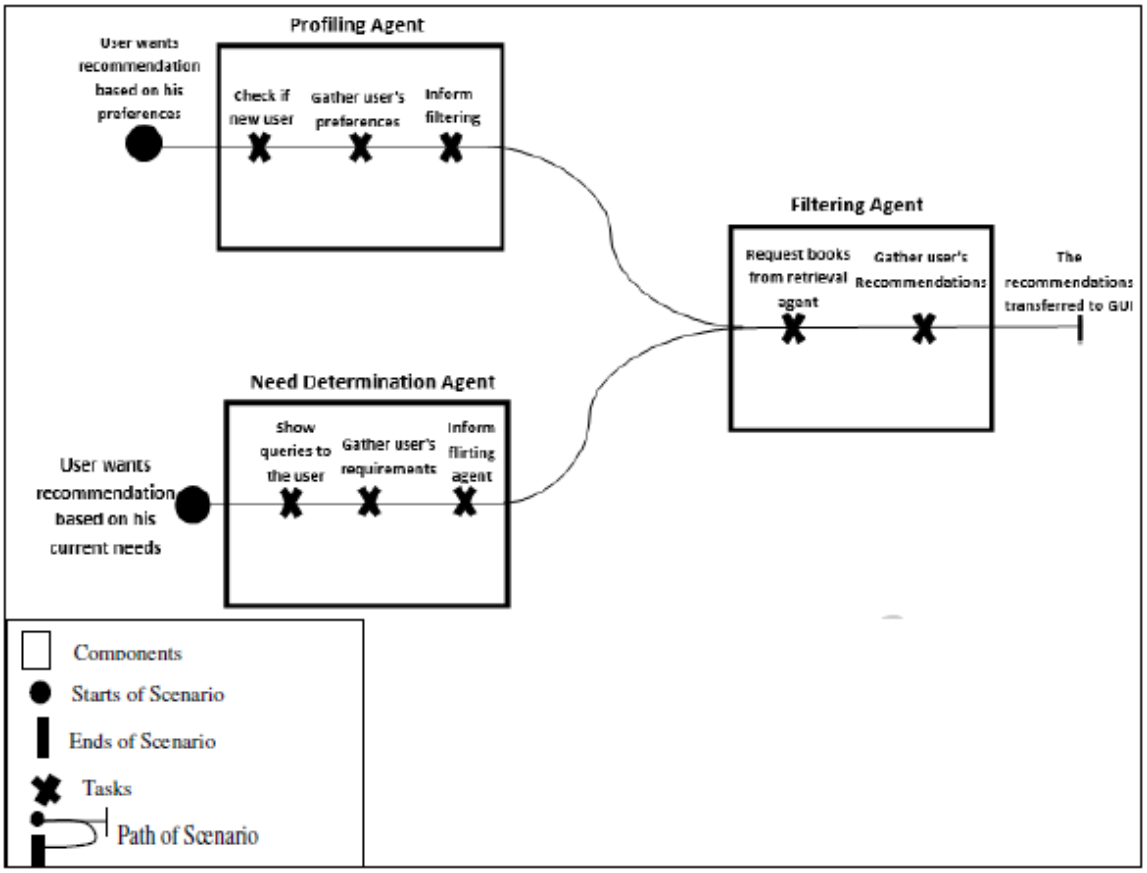


Figure 4: The UCM of translating book mechanism.

Step 2. If the system requirement specifications (SRS) [27] of a system do not have a translation function; then, this function is considered as Gold Plating concept; therefore, we should apply the FGA3 which avoid the part of gold plating represented in translation agent (TA) and all components connected from AD as illustrated in the figure below.

Depending on FGM1 the hierarchical decomposition approach (HDA) could be applied on books recommendation system to demonstrate the main components in visual manner to increase the understandability. Next table shows the main components and their connected components in books recommendations system.

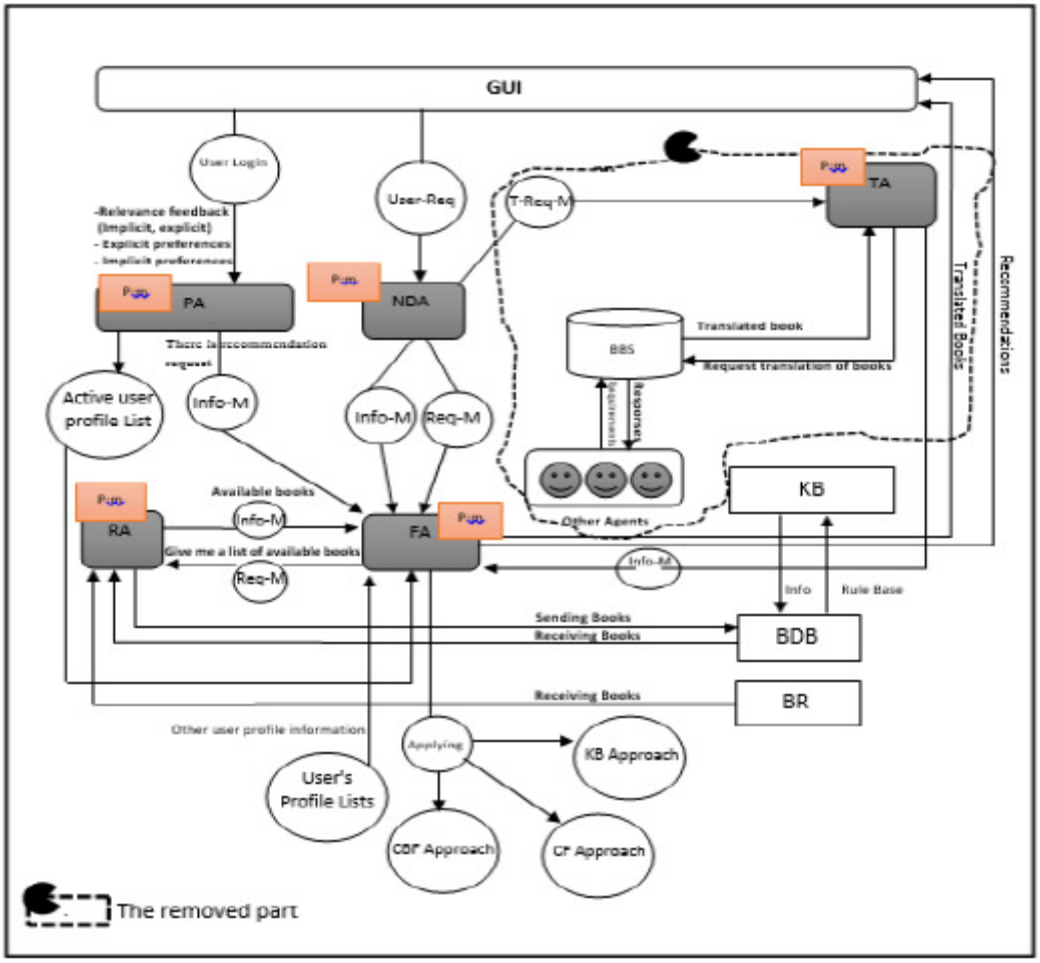


Figure 5: Omitting the part representing the gold plating

Table 3: The main components and their connected components in books recommendations

Main Components	**Connected component(1)**	**Connected component(2)**	**Connected component(3)**
Retrieval Agent	Book Data Base	Filtering Agent	Book Resource
Filtering Agent	Knowledge Base	GUI	Retrieval Agent
Profiling Agent	GUI	-	-
Need determination Agent	GUI	-	-
Book Data Base	Retrieval Agent	-	-
Book Resource	Retrieval Agent	-	-
Knowledge Base	Filtering Agent	-	-
GUI	Profiling Agent	NDA	Filtering Agent

Next figure demonstrates the majeure components in case study by applying HDA.

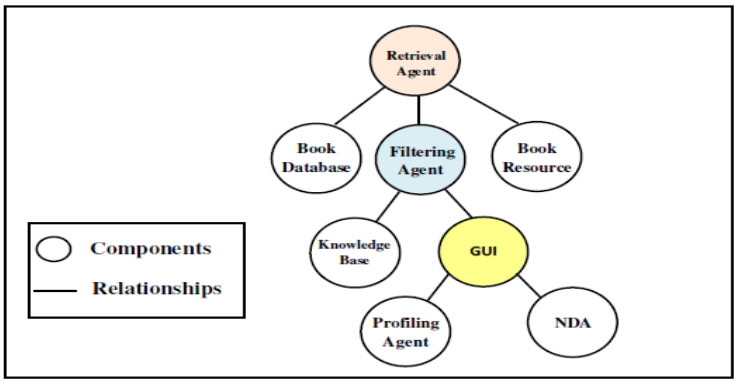


Figure 6: Conceptual system after applying HAD

Step 3. As we have pointed out, the modularity has a major role in decreasing the complexity in software design since the interaction among agents to accomplish their tasks can lead to system complexity. This step totally relies on cohesion measurement principle which uses the Communication Cohesion Measurement (CCM). This measurement works as a testing tool. This enables us to discover which agent needs more decompositions. In this research work, we have four agents described in the case study: filtering agent, profiling agent, need determination agent, and retrieval agent in respect that the translation agent has been omitted in the last step. The formulation of communication cohesive measurement is .The next illustration shows how.

Based on the architecture design of book recommendation system, the filtering agent has 4 internal relationships and 2 external relationships, profiling agent has just one internal relationship and 4 external relationships, need determination agent has one internal relationship and 2 external relationships and retrieval agent has 4 internal relationships and 3 external relationships as shown in the following:

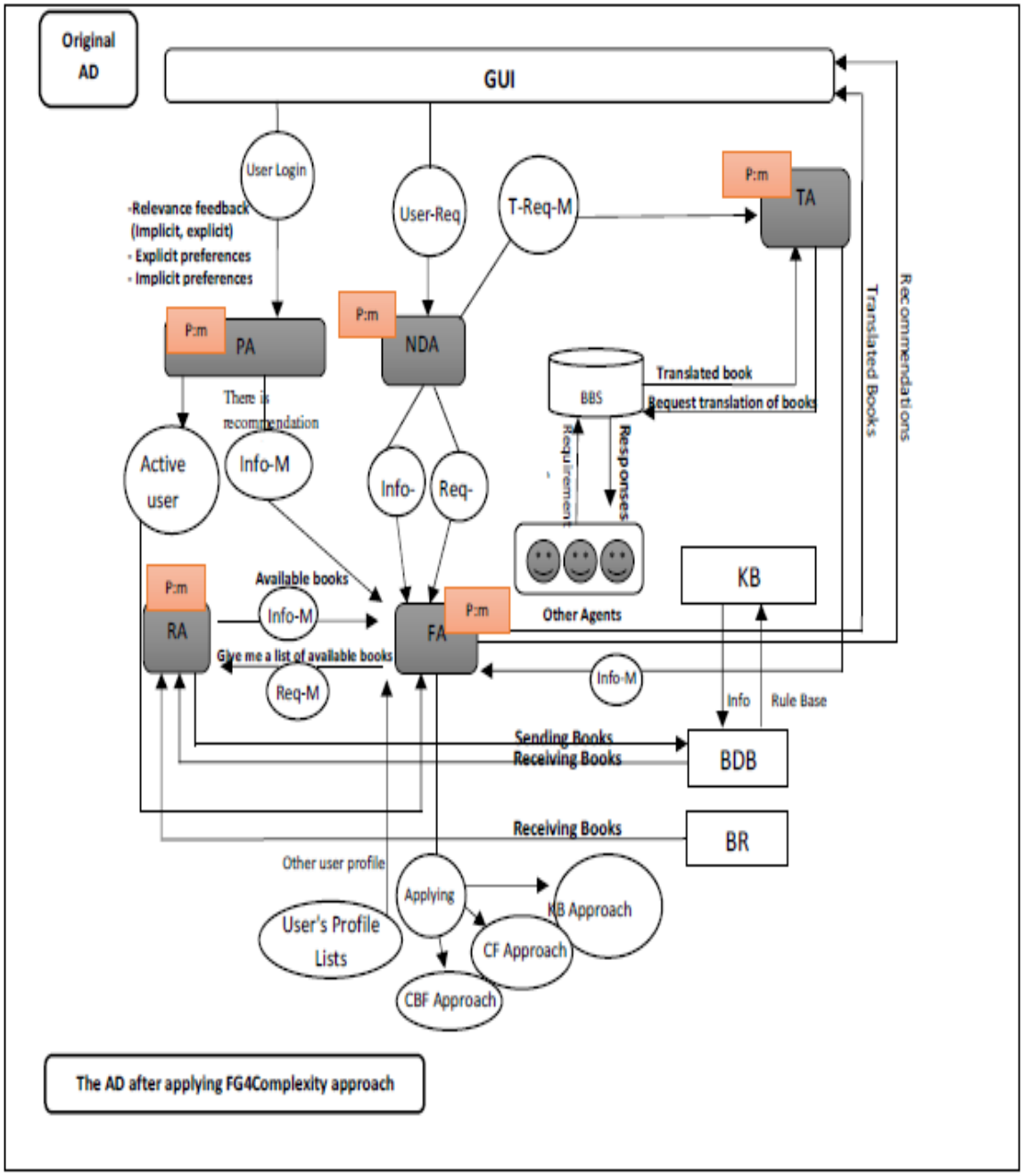
$$\mathbf{CCM\ (Ai)} = \frac{\mathbf{R}\ \text{internal}}{\mathbf{R}\ \text{internal} + \mathbf{R}\ \text{External}}$$

Table 4: The calculating by using CCM technique

Profiling agent		**Filtering agent**		**NDA**		**Retrieval agent**	
R internal	1	**R** internal	7	**R** internal	1	**R** internal	3
R external	2	**R** external	4	**R** external	3	**R** external	2
CCM(**PA**)	1/3	CCM(**FA**)	7/11	CCM(**NDA**)	1/4	CCM(**RA**)	3/5
Assessment		**Assessment**		**Assessment**		**Assessment**	
CCM(**PA**) = 0.3		CCM(**FA**) = 0.6		CCM(**NDA**) = 0.3		CCM(**RA**) = 0.6	

So, the results are:CCM (FA)<0.91, CCM(NDA) <0.91, CCM(RA) <0.91, and CCM(PA)<0.91. It is worth noticing that all results less than 0.91 by this, they do not need more decomposition.

Step 4. Applying a group of FG on the architecture design. This group consist of FGA1, FGA2, FGA3 and FGMOD2 which influence the architecture directly and the changes can clearly be observed.Next figures show the architectural design before and after applied **FG4Comlexity** approach.



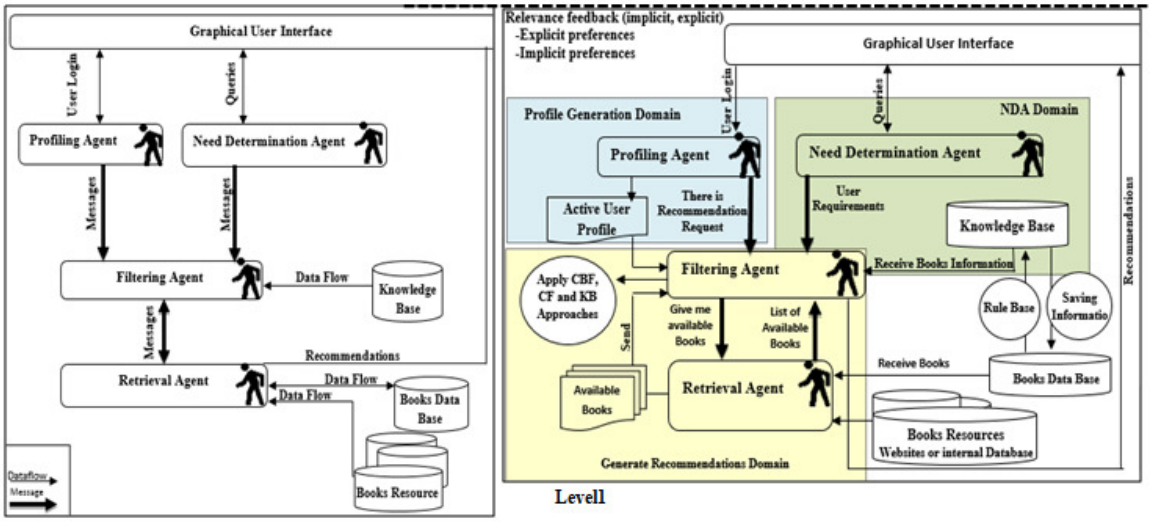


Figure 7: Displaying the AD before and after applying the FG4Complexity approach

4. CONCLUSION

The Research work approached the complexity of architectures design (AD) in systems based on multi agents (MAS) by a proposed solution method represented in a set of guidelines. These guidelines were introduced by extracting the factors affecting the complexity from three major sides of AD represented in abstraction, modularity and modeling thus, the approach labeled as "FG4complexity". It discussed the decrease of coupling which usually occurs during the interactions among agents and supporting the understandability of MAS architectures. The FG4complexity approach is useful for large systems such as recommendation systems that are based on MAS to avoid the complexity problems found in the most existing architectures. Thus, it enhances the quality standards, the reduction of complexity from (AD), and eventually reinforces the reusability concept.

FUTURE WORK

For future work, other aspects of architecture design will be addressed to attempt to make the proposed approach more effective. Those aspects may be are represented in the style, design patterns, documentation and so on. ALSO, we hope to apply the FG4complexity approach on other larger and more complex systems.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank ALLAH, without ALLAH this work would never have been finished.

I would like to express my sincere thanks to my supervisor Dr. Twfig Eltwel for his invaluable guidance and advice. I would like to thank my beloved husband Fathi El faitouri for his unlimited and faithful support as well as his patience and unconditional love. Also, The last but not least, I am profoundly grateful to my kind friend Asya Sohaim for her fruitful collaboration and advice. Finally, I thank everyone who encouraged me.

REFERENCES

[1] I. a. S. Markic, Maja and Maras, Josip, "Intelligent Multi Agent Systems for Decision Support in Insurance Industry," in Information and Communication Technology, Electronics and Microelectronics (MIPRO), ed: IEEE, 2014, pp. 1118--1123.

[2] M. Oprea, "Applications of multi-agent systems," in Information Technology, ed: Springer, 2004, pp. 239-270.

[3] Z. S. Ahmed Taki, "Formal Specification of Multi-Agent System Architecture," presented at the International Conference on Advanced Aspects of Software Engineering, ICAASE,, 2014.

[4] K. O. Chin, K. S. Gan, R. Alfred, P. Anthony, and D. Lukose, "Agent Architecture: An Overviews," Transactions on science and technology, vol. 1, pp. 18-35, 2014.

[5] S. A. D. Mark F. Wood, "An Overview of the Multiagent Systems Engineering Methodology," Springer, vol. 1957, 2001.

[6] B. H. Far, "Software Agents: Quality, Complexity and Uncertainty Issues," IEEE, 2002.

[7] B. R. Sinha, P. P. Dey, M. Amin, and H. Badkoobehi, "Software complexity measurement using multiple criteria," Journal of Computing Sciences in Colleges, vol. 28, pp. 155-162, 2013.

[8] D. N. M. Ghazal Keshavarz, Dr. Mirmohsen Pedram "Metric for Early Measurement of Software Complexity," International Journal on Computer Science and Engineering (IJCSE) vol. 3, 2011.

[9] M. Elammari and W. Lalonde, "An agent-oriented methodology: High-level and intermediate models," in Proc. of the 1st Int. Workshop. on Agent-Oriented Information Systems, 1999, pp. 1-16.

[10] M. E. T. Abdelaziz1, R. Unland3, C. Branki4, "MASD: Multi-Agent Systems Development Methodology," Multiagent and Grid Systems Journal,, 2010.

[11] S. a. D. Wagner, Florian, "Abstractness, Specificity, and Complexity in Software Design," ACM, pp. 35--42, 2011.

[12] F. Tsui, A. Gharaat, S. Duggins, and E. Jung, "Measuring Levels of Abstraction in Software Development," in SEKE, 2011, pp. 466-469.

[13] B. J. Kirandeep Kaur, Rekha Rani, "Analysis of Gold Plating: A Software Development Risk," International Journal of Computer Science and Communication Engineering, vol. 2, 2013.

[14] F. Medeiro, B. Pérez-Verdú, and A. Rodríguez-Vázquez, Top-down design of high-performance sigma-delta modulators vol. 480: Springer Science & Business Media, 2013.

[15] H. a. v. V. De Bruin, Hans, "Quality-driven software architecture composition," Journal of Systems and Software, Elsevier, vol. 66, pp. 269--284, 2003.

[16] S. Misra, "An approach for the empirical validation of software complexity measures," Acta Polytechnica Hungarica, vol. 8, pp. 141-160, 2011.

[17] A. Lawgali, "TRACEABILITY OF UNIFIED MODELING LANGUAGE DIAGRAMS FROM USE CASE MAPS," International Journal of Software Engineering & Applications (IJSEA), 2017.

[18] A. Zalewski, "Modelling and evaluation of software architectures," Prace Naukowe Politechniki Warszawskiej. Elektronika, 2013.

[19] B. L. M. Montaner, and J. De La, "A Taxonomy of Recommender Agents on the Internet," Artificial Intelligence Review, vol. 19, 2003.

[20] H. Castillo, "Hybrid Content-Based Collaborative-Filtering Music Recommendations," Department of Computer Science, Information System Engineering (ISE), Netherlands, 2007.

[21] J. Itmazi, "Flexible Learning Management System To Support Learning In The Traditional And Open Universities," PhD Thesis , university of Granada, 2005.

[22] J. Obando, "Methodology to obtain the user's Human Values Scale from Smart User Models," PhD Thesis, Department of Electronics, Computer Science and Automatic Control, University of Girona, 2008.

[23] R. Burke, "Hybrid Recommender Systems:Survey and Experiments," User Modeling and User-Adapted Interaction, vol. 12, 2002.

[24] T. T. a. R. Cohen, "Hybrid Recommender Systems for Electronic Commerce," in the 17th National Conference on Artificial Intelligence AAAI, 2000.

[25] R. N. A. A. E. frerjani, "Towards A General Architecture for Building Intelligent, Flexible, and Adaptable Recommender System Based on MAS Technology," post graduation, compluter science, benghazi IEEE journal, 2010.

[26] E. M. Saleh, "Architecture for Design Pattern Selection based on Multi-Agent System," post graduation, benghazi university, 2014.

[27] P. Thitisathienkul and N. Prompoon, "Quality assessment method for software requirements specifications based on document characteristics and its structure," in Trustworthy Systems and Their Applications (TSA), 2015 Second International Conference on, 2015, pp. 51-60.