Journal of Information Systems and Telecommunication
Vol.10, No.1, January-March 2022, 28-36
http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

An Agent Based Model for Developing Air Traffic Management Software

Mahdi Yousefzadeh Aghdam[1] ,Seyed Reza Kamel Tabbakh[2*],Seyed Javad Seyed Mahdavi Chabok[2] ,Maryam Kairabadi[1]

[1].Department of Computer Engineering, Neyshabur Branch, Islamic Azad University, Neyshabur, Iran
[2].Department of Computer Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran

Received: 23 Nov 2020/ Received: 04 Aug 2021/ Accepted: 03 Sep 2021
DOI: https://doi.org/10.52547/jist.15635.10.37.28

Abstract
The Air Traffic Management system is a complex issue that faces factors such as Aircraft Crash Prevention, air traffic controllers pressure, unpredictable weather conditions, flight emergency situations, airplane hijacking, and the need for autonomy on the fly. agent-based software engineering is a new aspect in software engineering that can provide autonomy. agent-based systems have some properties such: cooperation of agents with each other in order to meet their goals, autonomy in function, learning and Reliability that can be used for air traffic management systems. In this paper, we first study the agent-based software engineering and its methodologies, and then design a agent-based software model for air traffic management. The proposed model has five modules .this model is designed for aircraft ,air traffic control and navigations aids factors based on the Belief-Desire-Intention (BDI) architecture. The agent-based system was designed using the agent-tool under the multi-agent system engineering (MaSE) methodology, which was eventually developed by the agent-ATC toolkit. In this model, we consider agents for special occasions such as emergency flights' and hijacking airplanes in airport air traffic management areas which is why the accuracy of the work increased. It also made the flight's sequence arrangement in take-off and landing faster, which indicates a relative improvement in the parameters of the air traffic management

Keywords: Agent-Based Software Engineering; Agent-Based Modeling; BDI Architecture; Enterprise-Oriented Software Engineering; MaSE Methodology.

1- Introduction

Air traffic management system In addition to arranging and managing airspace controlled faces a complex distributed system with unpredictable cases that can disrupt the order of the ordinary stream. The decision to make optimal air traffic layout in such situations is matters to the routing independence and intelligent prevention of collision for each flight .agent-based software engineering provides facilities for designing and building complex distribution systems. In this methodology, the agent-oriented approach to designing distributed and complex air traffic management systems were studied. The system is faced with factors such as aircraft crash prevention, air traffic controllers pressure, unsuspended weather conditions, flight emergencies, airplane hijacking, and the need for autonomy on the fly. for this purpose, it is essential to develop new systems for air traffic flow [2]. The new approaches in the field of agent-based software engineering and Multi-agent Systems (MAS) have also brought us a new solution. These systems can simulate the air traffic control models based on the autonomy of operation and the relative autonomy of planes whether within the scope of airport control or in other areas in controlled space. In this paper, we first study agent-based software engineering and its methodologies, and then design an agent-based software model for air traffic management. The proposed model has five modules .this model is designed for aircraft, air traffic control and navigationsaids factors based on the belief- desire- Intention (BDI) architecture. The agent based system was designed using the Agent-Tool under the Multi-agent system engineering (MaSE) methodology, which was eventually developed by the agent- ATC toolkit. In this model, we consider agents for special occasions such as emergency flights' and hijacking airplanes in airport air traffic management areas which is why the accuracy of the work increased. It also made the flights' sequence arrangement in take-off and landing faster, which indicates a relative improvement in the parameters of the air traffic management.

✉ **Seyed Reza Kamel Tabbakh**
rezakamel@computer.org

The following sections of the paper have been structured, with Sect2. "Literature review" reviews the previous methods in the agent-based software of ATM field, Sect3. "BDI architecture" and sect4. "agent tool methodology design" describes the methodology of the agent system. In Sect5. "The MaSE- based analysis and design steps in agent tool" explain about how to Implementation of system. then in Sect6. "Analysis and design of the proposed model" implementation of the proposed model of air traffic management model and evaluates it; in addition, Sect. 8 has been dedicated to a conclusion.

2- Literature Review

The study of the behavior of air traffic management (ATM) systems using modeling systems and simulation tools can aid in the development and optimizing of new methods of improving ATM performance. The agent-based software system was applied as a solution for developing models that capture air traffic decisions and interactions with an adequate level of detail [29].agent-based models gained popularity as an effective approach to modeling and simulating specific dynamics of ATM systems. Different agent-based approaches were applied in ATM studies concerning air traffic control (ATC). These agent-based models represent air traffic processes such as vehicle trajectories, collision avoidance, airport operation, etc. In this category of models, we can mention Air MIDAS [30], ACES [31], AgentFly [32] and tools such as AirTop [33] and CAST [34]. These approaches are capable of simulating how air traffic controllers manage aircraft as they fly and move in the airports with a high level of detail. For example, Air MIDAS simulates the behavior of the final approach of aircraft in the terminal airspace and their interaction between the pilots and the flight controllers for risk evaluation. ATM studies related to air traffic flow management (ATFM) have recently been performed using also agent-based approaches. In general, they simulate ATFM on the day of the flight [35].

3- BDI Architecture

The BDI architecture or the belief-desire-intention (BDI) architecture is based on three concepts of belief, desire, and program. Belief is a statement of what the agent thinks about. BDI agents have a set of beliefs that are like the set of facts in law-based systems [4].

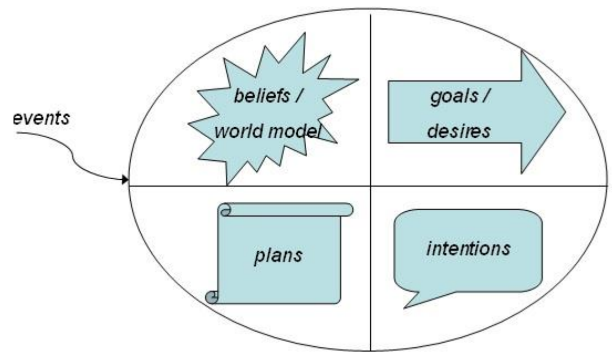

Fig 1. BDI Architecture Structure [4]

Tendency is the target state that the agent likes to achieve and the agent's plans are plans that tell the agent how to behave to achieve his or her intentions. An agent can have a program that triggers a specific activity. In this case, the agent will do exactly the same. Instead, the agent can have a program that can access a particular state. When a factor is committed to performing a specific activity or achieving a particular purpose, it promises to do so. Thus, the BDI will have a set of beliefs that will lead it to satisfy a set of tendencies. In order to achieve this, the BDI agent considers a set of options and chooses one or more of them. These options will now be operational [5,16].

4- Agent Tool Methodology Design

Today, agents usage has considerably increased in computer systems. objectification methods are not able to meet the needs of agent-based software and require essential equipment, which is why software engineering developed from object-oriented to agent-oriented [8]. Openness, high software complexity, distributed data sources and control, and high flexibility are features of systems that the agent technology allows them to generate. The program also has the highest quality and productivity and has the lowest cost [10]. The role of software engineering is to provide models and techniques that facilitate the production and maintenance of the software. with the emergence of a new perspective on the production of software systems, programming languages, software tools and software engineering methods are also appropriate. factors are very similar to objects, but the view and characteristics that distinguish one factor from an object do not allow object-oriented methods to operate on systems based on the factor. For this reason, these systems require software engineering methods based on the unique features of the agents [9,11,12]. Various methodologies have been developed for analyzing and designing agent-based systems [14]. These methodologies can be categorized into two major categories. The first is the methodologies that are based on the development of object-oriented software engineering techniques and their adaptation to the agent's perspective. These methodologies are Gaia, multi-agent system engineering: MaSE [16,17]. MESSAGE [13]. The second category is the methodology that develops engineering knowledge methods.MAS-CommonKADS [18]. CoMoMAS [19] are examples of these methodologies.

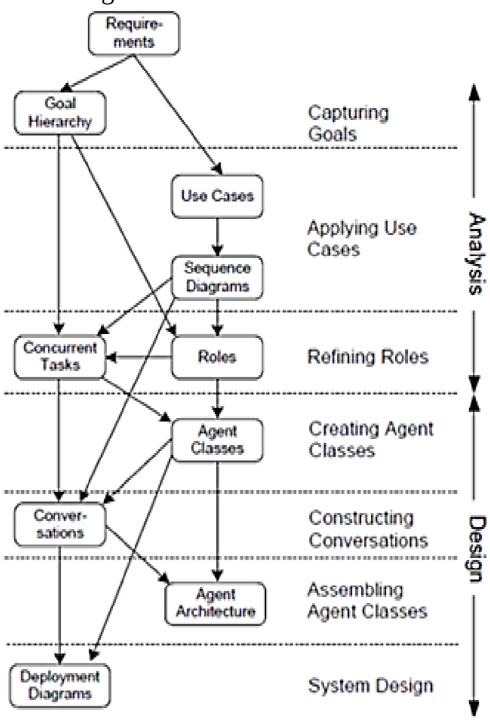

Fig 2. Design and analysis steps in the multi-agent system engineering: MaSE methodology [28]

MaSE is one of the software engineering methodologies for the operating system based on the operating system [10]. Which, in seven steps, passes through the stages of analysis and design, the software engineer implements the problem-to-implementation definition [16, 17]. With the support of the agent-Tool tool that covers all stages of analysis and design and enables the production of code automatically [16, 17]. this methodology is considered one of the most important methodologies in the software engineering of agent-based systems. The MaSE methodology has been used in the fabrication and production of various systems, as well as various studies to address its shortcomings [22, 21,20,17].

5- The MaSE-based Analysis and Design Steps in Agent Tool

5-1- System Analysis Steps

The MaSE methodology consists of two main phases of analysis and design. the MaSE analysis phase consists of three main phases, "setting goals", "defining the applications", and revision of roles in the system [21] ,and the design phase related to topics such as the diagram and classroom conversation [10]. The design and analysis steps in this method are shown in (figure2). [23,22]

5-2- Setting Goals

The first stage of analysis in MaSE is the system's "goal setting" stage, which itself consists of two steps: "detecting targets" and determining the hierarchical chart of objectives. At the stage of diagnosing the objectives, using the definition of the system's requirements, the basic objectives of the system are identified. The "detection of targets" stage begins with the creation of a general scenario of the system. then, based on this scenario, the overall goals of the system are determined. In the stage of classification of goals, the goals set for the system are categorized hierarchically in the form of a tree.

5-3- Determine the Application Cases

The purpose of this step is to identify a set of roles in the system and how they communicate. The "determination of its applications" phase consists of two steps, the creation of "application cases" and the creation of "order charts." In the process of creating "applications", the set of events that may occur in the system is defined. In the process of constructing sequence diagrams, based on the application cases created in the previous step, the order of events between the various roles is displayed using the "order chart".

5-4- Role Reviewing Step

"Role Revision" is the last stage of analysis in MaSE. The purpose of this step is to determine the final roles defined in the system appropriately for design and implementation in a multiprocessor system. At this stage, after the roles are identified, the "tasks" of each role are determined [20]. Each task is displayed using an infinite mode machine. In this case, the "Order Diagram" can be used as the starting point for specifying these state machines, which, in turn, can express the desired states in plain language, provided they are not ambiguous.

5-5- System Design Steps

The models created at the analysis stage are the input of the design stage at MaSE. The operating classes, the effects of each "conversation" on each factor, the internal architecture of the agents, and the "structural arrangement" of the software output system are the design stage [20].

5-6- Create Operating Classes

The first step in designing is the creation of operating classes. At this stage, operating classes are created based on the roles defined in the analysis stage. The agents have different roles in the process of running the system.

5-7- Stage of Dialogue Creation

At this stage, for each connection, a finite state machine is drawn up for each of the agents involved in that connection. Messages in this section are defined based on UML contracts.

5-8- Step Combine Operating Classes

At this stage, the internal architecture of the agents is determined. The internal classes of each type of agent are determined by a set of predefined components and by their combination [24]. The "reactive" agents have the response to the stimulus to the environment. Architectures based on this are based on specific rules. (figure 2)

5-9- Stage System Design

The final stage in MaSE is the system design. At this stage, the actual examples of each agent in that system are displayed in the "structural arrangement" system diagram. The number, type, and location of each sample created in this section are indicated.

6- Analysis and Design of the Proposed Model

The agents are very similar to objects, but the view and characteristics that distinguish one factor from an object do not allow object-oriented methodologies to be appropriate for in-system systems. For this reason, these systems require software engineering methods based on the unique features of the agents [13,12]. So far, various methodologies have been developed for analyzing and designing agent-based systems [14]. Air traffic has steadily increased over the past decade, and ICAO has predicted that annual traffic will double in 2030 compared to 2013. [25]. Air traffic controllers are responsible for ensuring that the flights' are safe and smooth, and if flights' are diverted from the main route, they should be able to quickly return to the main path and to do this, routes between each airport should be maintained [26].

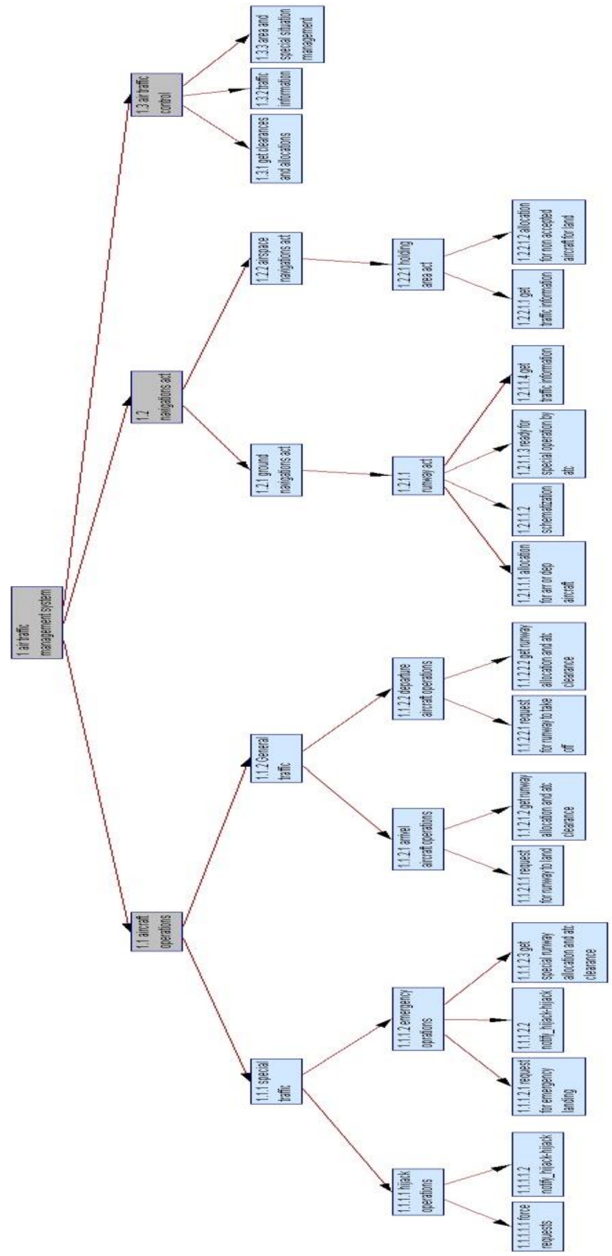

Fig 3. the hierarchical representation of the objectives of the multi-agent system of air traffic control management

The air traffic system in a network has vertices that represent the geographic location that the airplane can cross over. [27] The goals of the multistage air traffic control system are based on the air traffic control scenario and optimal traffic arrangement and avoidance of any incident. On the other hand, these goals should be in such a way as to satisfy the demands of a system designed in the form of artificial intelligence technology based on multitasking systems, which is to solve the existing problem and not complex the problem. As explained in the previous section, the objectives of our proposed model are categorized according to the MaSE methodology and are presented as a hierarchical chart of objectives [20].

6-1- Hierarchical Chart of Goals

These goals are organized in three general categories, which outline the objectives of the operating groups in the system and the objectives of the system. It should be noted that in determining the objectives, the status of the system should be considered collectively or competitively, because in a multi-competitive system, there is usually a conflict between the goals of the agents and the objectives of the entire conflict system, and the profit of one factor of the system alone contrasts with the total profit of the system, but in collaborative multitasking systems, this is rare. Here, our system is a collaborative multimodal system, since its main objective is aeronautics and air
traffic control. The ultimate goal in controlling air traffic control is to optimize the arrangement of air traffic in the airspace and to avoid any disaster. But a multi-agent system has been created from a variety of factors, each with a specific activity and purpose. Each factor, while playing its role in the system, is actually looking for their goal, which we will achieve in the final conclusion of the activities and goals to achieve the main purpose of the system.
Given the three factors in our intelligent proposed model for controlling Air traffic the main purpose of the model is divided into three main subdivisions: aviation factor group goals, airborne mission group goals, atc task force objectives. For example, the objectives of the aircraft group are divided into two sub-divisions of the objectives of ordinary aircraft and the objectives of the special aircraft (at risk). Objectives of the common aircraft can be divided into two sub-groups of the objectives of the incoming aircraft and the objectives of the operating aircraft of the outboard, and then the objectives of the operating aircraft of the aircraft are divided into two categories: the allocation of the band for landing at the airport and the assignment of the band and the request for authorization of the ATC agent landing in the airport is divided according to the priority of the flight. In (figure3), is presented the hierarchical chart of the objectives multi-factor traffic control system.

6-2- System Sequences

In this section, we will detail the five sequence diagrams that result from the objectives and system operation.

6-2-1- Arrangement Diagram of the Incoming Airplane Landing Operation

The incoming air carrier will send its flight information to the operating tower of the flight tower to assign the landing gear, as well as the incoming airline agents in the aircraft operating group to send their flight information. Upon receipt by the ATC agent, the agent sends a request for the occupation of the airport landing bands to the carrier's airborne commander, and then the agent sends the request to determine the best condition for the sub-category landing bands which it sends itself. After identifying the best landing route, it sends its specifications to the ATC agent, and the agent sends the updated timetable to the ATC agent, after sending the assigned command to the target band. The ATC agent sends permission to assign the specified band to the intended aircraft. (figure. 4)

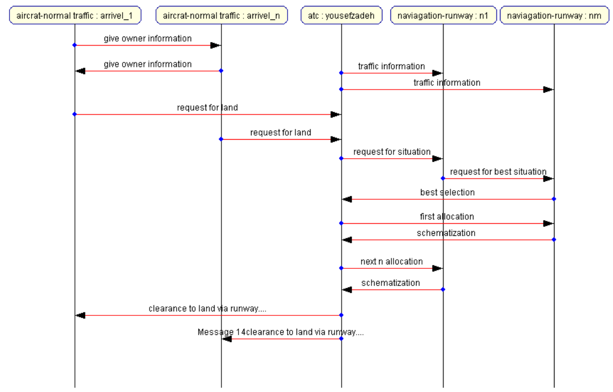

Fig 4. Arrival chart of the incoming airplane landing operation

6-2-2- Arrangement Diagram Outboard Aircraft Lifting Operation

The outboard agent sends a request to the flying tower to allocate the exit channel to the flight attendant. Also, the airliner sends flight information to the aircraft operating group. Upon receipt by the ATC agent, this agent sends a request for the occupation time of the airport take-off bands from the airport's airborne group, and then each of them reports their status (declaring readiness or being busy) to the agent ATC sends.
The ATC agent sends the assignment command to the desired band. The operating agent sends its updated timetable to the ATC agent. This agent sends the necessary permission to assign the specified band to the agent (aircraft) concerned. Of course, it should be noted that the ATC agent sends traffic information to the navigation agents in order to accelerate the system's performance. (figure.5)

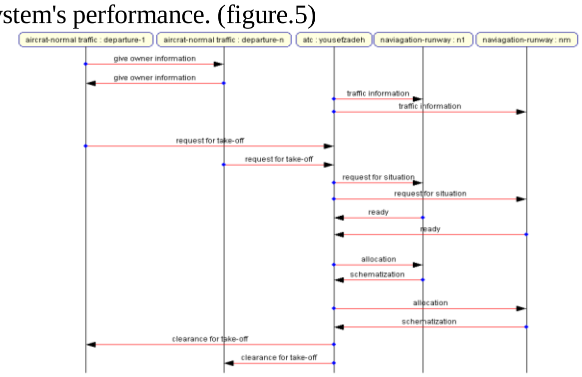

Fig 5. Arrangement diagram of the lifting operation of the exits

6-2-3- Arrangement Diagram of the Operation of Entering the Aircraft into the Holding Area

The incoming air carrier will send its flight to the tower of the flight tower in order to allocate the landing gear. Upon receipt by ATC, this agent sends an application to the occupant of the airport landing bands to send the airline flight officer headquarters. The director of the navigational group after the survey and during the steps mentioned above, sends an answer to the fact that all the airport landing bands are busy with the ATC agent. The ATC agent then sends the agent (aircraft) a non-landing message. The ATC agent sends the flight traffic information to the navigation operator of the landing area, and then issues the decree to the operating agent (aircraft) and the license to start the hiding area project. (figure. 6)

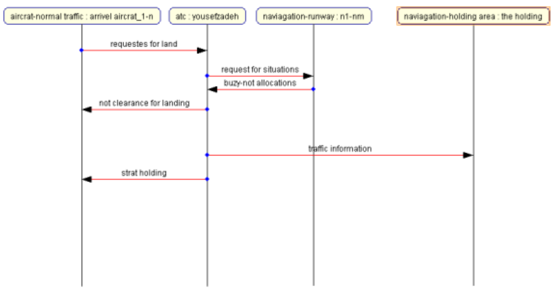

Fig 6. Airplane Entry Operations to the Holding Area

6-2-4- Flow Diagram Diagrams in Emergency Situations

The agent also receives flight information from other aircraft agents. The ATC agent sends the traffic status to the airborne agents, and then sends a message to the emergency agent and alert mode to the navigation agent. After that, the ATC agent gives all the planes a decree decommissioning from the space specified for the emergency landing. (figure.7)

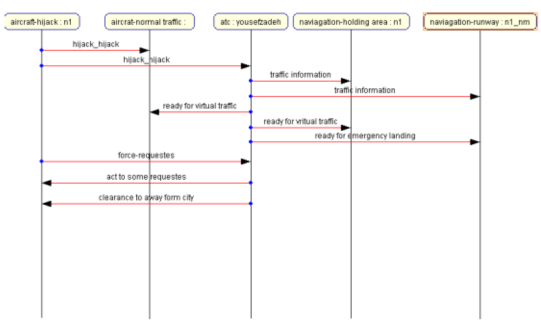

Fig 7. Operations Diagram of Emergency Operations

6-2-5- Arrangements for Aircraft Operations During the Hijacking

The airplane hijacking agent's airborne agent will send the flight agent to the flying tower and other flying agents (onboard aircraft). The ATC agent sends the traffic status to airborne agents, and then sends a hailing message and requesting an alert status to the agents, including the landing gear and the holding area. The ATC agent then orders all planes to be ready for artificial traffic creation to limit the abducted airplane's space. The abducted aircraft agent sends requests to the ATC agent, and the ATC agent tries to meet their apparent needs by helping them from the virtual traffic mode to the abducted aircraft, sending messages that control and dispose of them from the urban and rural spaces. (figure.8)

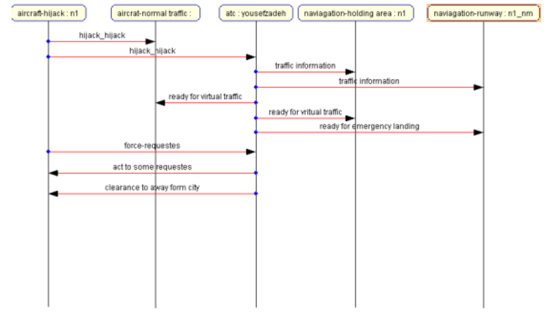

Fig 8. Arrivals of airplane operations during the hijacking

6-3- Graph of the Expression of Roles and Tasks

As described in Section 6-2, the "review roles" stage is the most important stage in MaSE's analysis. The goal of this step is to determine the roles defined in the system appropriately for design and implementation in a multi-purpose system. In the proposed model for air traffic management multifunctional system, according to classified objectives, several roles such as normal-type aircraft (inertial, outbound, local), abducted aircraft, emergency planes, flight controllers, flight bands (landing, take off), stops and so on. At this stage, after the roles are identified, the "tasks" of each role are determined. Given the sequence diagrams described in the previous section, which are described in detail, after the determination of the roles, it is time to determine the tasks. In our proposed model for the air traffic control multitasking system, tasks are divided into two categories: data processing (traffic and airport information) and flight instructions (requests, commands, issuance of licenses, etc.). "The role of flight attendance" also has a supervisory and supervisory role. Each of these roles includes and follows the specific objectives described in the hierarchical objectives. (figure.9)

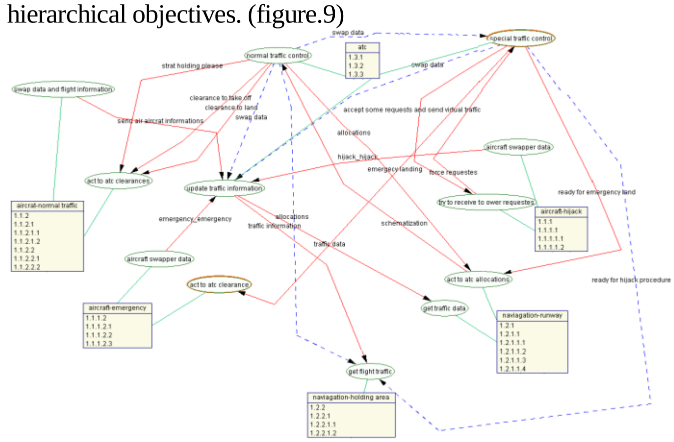

Fig 9. Graphs of Expressions of Roles and Function

6-4- Charts for Designing Operating Classes and their Combining Expressions

At this stage, operating classes are created based on the roles defined in the analysis stage. Factors play different roles during the process of system execution. In our proposed model for air traffic control management, based on the criteria mentioned, three functional class groups are considered, which include the operating group of flight care, the aircraft operating group, and the aeronautical operating group. The factor group is the facilitator of the operation mini the "reactive" factors have the characteristics of the stimulus- response to the environment. Architectures based on this are based on specific rules. (Fig.10)

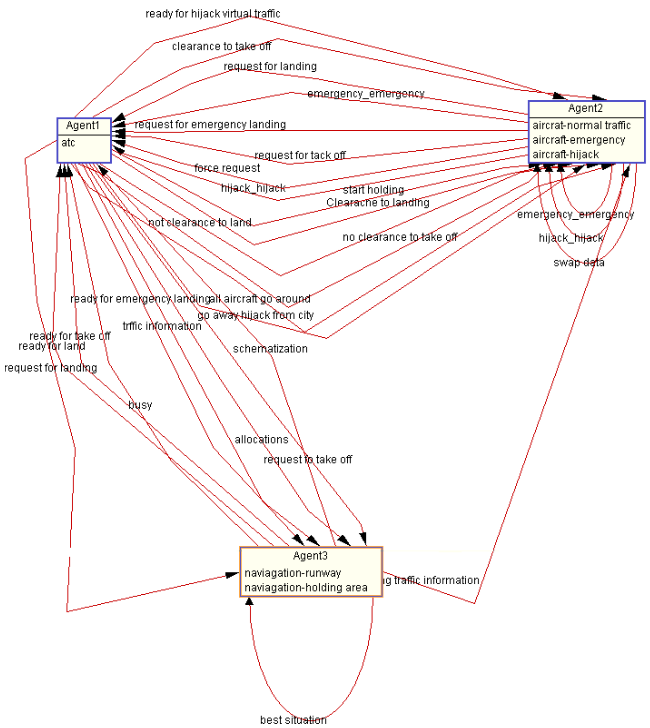

Fig 10. Design Diagrams of the Operating Classes and Expressions of Their Combinations

7- Discusion and Results

The goals have been regulated in the form of three sub-collection which are indications of the agent groups' goals in system and system goals. To determine the goals ,the situation of the system should be cooperative and or competitive, because ,in competitive multi-agent systems, there is a contradiction between agents goals and total goals system and the interest of an agent out of system contradicts with its total interest, but in a multi-agent system this problem is rare .

Here, our system has been determined with the kind of cooperative multi-agent systems, because our main goal is in the direction of stall speed and secure air traffic control.

On the basis of three agent bands in the suggested intelligent model to control air traffic, the major goal of the model is divided into three sub-classifications:

- aircraft agent group goals
- air navigational agent group goals
- ATC agent group goals

For example, aircraft agent group goal with two sub-structure:

- normal aircraft agent goals
- special agents goals(in risk)

also, normal agents goals are as follows:

- arrival (aircraft)agent goals
- departure(aircraft)agent goal

agents goals arrival (aircraft) are two sub-groups:

- request of allocating a runway to land in the airport

The runway allocation and request of license from ATC agent to land in the airport on the basis of flight priorities .Table 1 shows an example of flight priority rules derived from intelligent operating systems that help evaluate multi-agent consulting systems in air traffic management. Due to the increase of conditions and their combination, we will present some of the rules as an example in this table.

Table 1. flight rules in agent-base system

1- If aircraft arrival or departure from runway then aircraft movement into the wind
2- arrival aircraft has high preference than departure aircraft
3- If aircraft arrival into the holding area then { If aircraft input from part 1 then aircraft turn left and proceed to procedure "a" of holding area If aircraft input from part 2 then aircraft turn right till radial 30 degree and proceed to procedure "b" of holding area If aircraft input from part 3 then aircraft turn right and proceed to procedure "c" of holding area}
4- in arrival or departure if medium aircraft behind heavy aircraft then 2 minute (minima time separation) shall be applied
5- in arrival or departure if light aircraft behind medium or heavy aircraft then 3 minute (minima time separation) shall be applied
6- fighter aircraft same as light aircraft in priority.
7- hijack and emergncy must be away from city.
8- if(aircraft[i][3].compareTo("arr")==0;
9- if(aircraft[i][1].compareTo("emergency")==0 aircraf[i][6]=5;
10- if(aircraft[i][1].compareTo("air-amb")==0 aircraf[i][6]=4;
11- if(aircraft[i][1].compareTo("sar")==0 aircraf[i][6]=3;
12- if(aircraft[i][1].compareTo("vip")==0 aircraf[i][6]=2;
14- if(aircraft[i][3].compareTo("dep")==0;
13- if(aircraft[i][1].compareTo("acft")==0 aircraf[i][6]=1;
15- if(aircraft[i][1].compareTo("scramble")==0 aircraf[i][6]=5;
16- if(aircraft[i][1].compareTo("air-amb")==0 aircraf[i][6]=4;
17- if(aircraft[i][1].compareTo("sm")==0 aircraf[i][6]=3;
18- if(aircraft[i][1].compareTo("sar")==0 aircraf[i][6]=2;
19- if(aircraft[i][1].compareTo("vip")==0 aircraf[i][6]=1;
20- if(aircraft[i][1].compareTo("normal")==0 aircraf[i][6]=0;
21- if(aircraft[i][2].compareTo("heavy")==0 aircraf[i][7]=0.3;
22- if(aircraft[i][2].compareTo("medium")==0 aircraf[i][7]=0.2;
23- if(aircraft[i][2].compareTo("light")==0 aircraf[i][7]=0.1;

8- Conclusion

The use of intelligent agent-based systems in air traffic management is one of the important issues in the field of artificial intelligence knowledge and modern aeronautical sciences. In this paper, we tried to design a new model for a multi-agent traffic management system. In this model, the structure of the agents and their interactions are selected based on the BDI model. Then the design of the multi-agent system and operating classes under the MaSE methodology was done using the agent tools. Due to the fact that there was a distinction between emergency and airlift control and other flights', the results of the system test showed a relative improvement in the parameters being evaluated. In further studies, to use deep learning networks methods such as recurrent neural networks (RNN) and long-short term memory(LSTM) in agent-based air traffic model design to increase the ATM accuracy

References

[1] F. Zambonelli, N. R. Jennings, and M. Wooldridge, "Organisational abstractions for the analysis and design of multi-agent systems," in International Workshop on Agent-Oriented Software Engineering, 2000, pp. 235–251.

[2] ICAO, "Manual on Air Traffic Management System Requirements (doc. 9882)." p. 67, 2008

[3] V. N. Drozdov, V. A. Kim, and L. B. Lazebnik, "[Modern approach to the prevention and treatment of NSAID-gastropathy].," Ėksperimental'naia i klinicheskaia gastroėnterologiia = Experimental & clinical gastroenterology, no. 2. pp. 106–110, 2011.

[4] S. Zilberstein, "Book Review: 'Multiagent Systems: a Modern Approach To Distributed Artificial Intelligence', Gerhard Weiss," International Journal of Computational Intelligence and Applications, vol. 01, no. 03. pp. 331–334, 2001, doi: 10.1142/s1469026801000159.

[5] K. S. Obhan, "Multi-agent Systems: A Modern Approach to Distributed Artificial Intelligence By Gerhard Weiss.

[6] V. Smirnova, "Multi-agent System for Distributed Data Fusion in Peer-to-Peer Environment," Masters Study, 2002.

[7] M. Niazi and A. Hussain, "Agent-based computing from multi-agent systems to agent-based models: A visual survey," Scientometrics, vol. 89, no. 2, pp. 479–499, 2011, doi: 10.1007/s11192-011-0468-9.

[8] B. Henderson-sellers, "Agent-based Software Development Methodologies What is an Agent?" 1997.

[9] N. R. Jennings, "Agent-oriented software engineering," in European Workshop on Modelling Autonomous Agents in a Multi-Agent World, 1999, pp. 1–7.

[10] E. Amini and F. S. Gharehchopogh, "Analysis and Design of New Secure Dynamic Structure for Increasing Modularity of ERP Systems."

[11] N. R. Jennings, "On agent-based software engineering," Artif. Intell., vol. 117, no. 2, pp. 277–296, 2000.

[12] F. Zambonelli, N. R. Jennings, and M. Wooldridge, "Developing multiagent systems: The Gaia methodology," ACM Trans. Softw. Eng. Methodol., vol. 12, no. 3, pp. 317–370, 2003.

[13] R. Evans et al., "MESSAGE: Methodology for engineering systems of software agents," Eurescom, Edin, no. September 2001. pp. 223–907, 2001.

[14] C. A. Iglesias, M. Garijo, and J. C. González, "A survey of agent-oriented methodologies," in International Workshop on Agent Theories, Architectures, and Languages, 1998, pp. 317–330.

[15] A. M. de Souza, C. A. R. L. Brennand, R. S. Yokoyama, E. A. Donato, E. R. M. Madeira, and L. A. Villas, "Traffic management systems: A classification, review, challenges, and future perspectives," Int. J. Distrib. Sens. Networks, vol. 13, no. 4, 2017, doi: 10.1177/1550147716683612.

[16] M. O. Khozium, "Multi-agent system overview: architectural designing using practical approach," Int. J. Comput. Technol., vol. 5, no. 2, pp. 85–93, 2013.

[17] N. Glaser, State of the Art: Knowledge Acquisition and Modelling. 2002.

[18] G. Schreiber, B. Wielinga, R. de Hoog, H. Akkermans, and W. de Velde, "CommonKADS: A comprehensive methodology for KBS development," IEEE Expert, vol. 9, no. 6, pp. 28–37, 1994.

[19] N. Glaser, "Contribution to knowledge acquisition and modeling in a multi-agent framework PhD Thesis," UHP Nancy I, pp. 87–127, 1996.

[20] R. H. Bordini et al., "A survey of programming languages and platforms for multi-agent systems," Inform., vol. 30, no. 1, pp. 33–44, 2006.

[21] L. Bianco, P. Dell'Olmo, and A. R. Odoni, New concepts and methods in air traffic management. Springer Science & Business Media, 2013.

[22] International Civil Aviation Organization, "Global Air Navigation Plan for CNS / ATM Systems," Training, no. Doc 7300. 2002.

[23] J. C. Hill, F. R. Johnson, J. K. Archibald, R. L. Frost, and W. C. Stirling, "A cooperative multi-agent approach to free flight," Proceedings of the International Conference on Autonomous Agents. pp. 1205–1212, 2005, doi: 10.1145/1082473.1082637.

[24] D. Robinson, "A Component Based Approach to Agent Specification, Department of electrical and computer engineering Air Force Institute of Technology," M. Sc. Thesis, 2000.

[25] R. Breil et al., "Multi-agent Systems for Air Traffic Conflicts Resolution by Local Speed Regulation To cite this version: HAL Id: hal-01342623 Multi-agent Systems for Air Tra ffi c Conflicts Resolution by Local Speed Regulation," 2016.

[26] J. M. Canino, J. B. Portas, J. M. Molina, J. Garc\'\ia, and T. Magister, "A multi-agent approach for designing next generation of air traffic systems ," in Advances in Air Navigation Services , InTech, 2012.

[27] D. R. Isaacson, A. V Sadovsky, and D. Davis, "Tactical scheduling for precision air traffic operations: Past research and current problems," J. Aerosp. Inf. Syst., vol. 11, no. 4, pp. 234–257, 2014.

[28] Self, Athie L., and Scott A. DeLoach. "Designing and specifying mobility within the multiagent systems engineering ethodology." Proceedings of the 2003 ACM symposium on Applied computing. 2003.

[29] Molina, Martin, Sergio Carrasco, and Jorge Martin. "Agent-based modeling and simulation for the design of the future european air traffic management system: The experience of

cassiopeia." International Conference on Practical Applications of Agents and Multi-Agent Systems. Springer, Cham, 2014.

[30] Gore, Brian F. "Man–machine integration design and analysis system (MIDAS) v5: Augmentations, motivations, and directions for aeronautics applications." Human modelling in assisted transportation. Springer, Milano, 2011. 43-54.

[32] Šišlák, David, et al. "AGENTFLY: Towards multi-agent technology in free flight air traffic control." Defence industry applications of autonomous agents and multi-agent systems. Birkhäuser Basel, 2007. 73-96.

[33] Molina, Martin, Sergio Carrasco, and Jorge Martin. "Agent-based modeling and simulation for the design of the future european air traffic management system: The experience of cassiopeia." International Conference on Practical Applications of Agents and Multi-Agent Systems. Springer, Cham, 2014.

[34] Becker-Asano, Christian, et al. "A multi-agent system based on unity 4 for virtual perception and wayfinding." Transportation Research Procedia 2 (2014): 452-455.

[35] Bazzan, Ana LC, and Franziska Klügl. Multi-agent systems for traffic and transportation engineering. IGI Global, 2009.

[36] Breil, Romaric, et al. "Multi-agent systems to help managing air traffic structure." Journal of Aerospace Operations 5.1-2 (2017): 119-148.

[37] Abar, Sameera, et al. "Agent Based Modelling and Simulation tools: A review of the state-of-art software." Computer Science Review 24 (2017): 13-33.

[38] Gurtner, Gérald, et al. "An empirically grounded agent based simulator for the air traffic management in the SESAR scenario." Journal of Air Transport Management 59 (2017): 26-43.

[39] Bouarfa, Soufiane, Jasper Müller, and Henk Blom. "Evaluation of a Multi-Agent System approach to airline disruption management." Journal of Air Transport Management 71 (2018): 108-118